
MMPose

Release 0.29.0

MMPose Authors

Apr 06, 2023

CONTENTS

1	Prerequisites	3
2	Installation	5
2.1	Best Practices	5
2.2	Verify the installation	5
2.3	Customize Installation	6
2.4	Trouble shooting	8
3	Get Started	9
3.1	Prepare Datasets	9
3.2	Inference with Pre-trained Models	10
3.3	Train a model	12
3.4	Benchmark	14
3.5	Tutorials	15
4	Demo	17
4.1	2D Animal Pose Demo	17
4.2	2D Face Keypoint Demo	20
4.3	2D Hand Keypoint Demo	22
4.4	2D Human Pose Demo	24
4.5	2D Pose Tracking Demo	28
4.6	2D Human Whole-Body Pose Demo	31
4.7	3D Mesh Demo	34
4.8	3D Hand Demo	34
4.9	3D Human Pose Demo	35
4.10	3D Multiview Human Pose Demo	37
4.11	Hand Gesture Recognition Demo	39
4.12	Webcam Demo	39
5	Benchmark	43
5.1	Comparison Rules	43
5.2	Hardware	43
5.3	Software Environment	43
6	Inference Speed	45
6.1	Comparison Rules	45
6.2	Model complexity information and inference speed results of major models in MMPose	46
7	Overview	47
7.1	2D Animal Keypoint Dataset	47
7.2	2D Body Keypoint Datasets	47

7.3	2D Face Keypoint Datasets	48
7.4	2D Fashion Landmark Dataset	48
7.5	2D Hand Keypoint Datasets	48
7.6	2D Hand Keypoint Datasets	48
7.7	2D Wholebody Keypoint Datasets	49
7.8	3D Body Keypoint Datasets	49
7.9	3D Body Mesh Recovery Datasets	49
7.10	3D Hand Keypoint Datasets	49
8	2D Body Keypoint Datasets	51
8.1	COCO	51
8.2	MPII	52
8.3	MPII-TRB	53
8.4	AIC	54
8.5	CrowdPose	54
8.6	OCHuman	55
8.7	MHP	56
8.8	PoseTrack18	57
8.9	sub-JHMDB dataset	58
9	2D Wholebody Keypoint Datasets	61
9.1	COCO-WholeBody	61
9.2	Halpe	62
10	2D Face Keypoint Datasets	65
10.1	300W Dataset	65
10.2	WFLW Dataset	66
10.3	AFLW Dataset	67
10.4	COFW Dataset	68
10.5	COCO-WholeBody (Face)	69
11	2D Hand Keypoint Datasets	71
11.1	OneHand10K	71
11.2	FreiHAND Dataset	72
11.3	CMU Panoptic HandDB	73
11.4	InterHand2.6M	74
11.5	RHD Dataset	74
11.6	COCO-WholeBody (Hand)	75
12	2D Fashion Landmark Dataset	77
12.1	DeepFashion (Fashion Landmark Detection, FLD)	77
12.2	DeepFashion2	78
13	2D Animal Keypoint Dataset	81
13.1	Animal-Pose	81
13.2	AP-10K	83
13.3	Horse-10	84
13.4	MacaquePose	84
13.5	Vinegar Fly	85
13.6	Desert Locust	86
13.7	Grévy's Zebra	87
13.8	ATRW	87
14	3D Body Keypoint Datasets	89
14.1	Human3.6M	89

14.2	CMU Panoptic	90
14.3	Campus and Shelf	91
15	3D Body Mesh Recovery Datasets	93
15.1	Notes	93
15.2	COCO	94
15.3	Human3.6M	95
15.4	MPI-INF-3DHP	96
15.5	LSP	97
15.6	LSPET	97
15.7	CMU MoShed Data	98
16	3D Hand Keypoint Datasets	99
16.1	InterHand2.6M	99
17	Overview	101
17.1	Animal	101
17.2	Body(2D,Kpt,Sview,Img)	102
17.3	Body(2D,Kpt,Sview,Vid)	105
17.4	Body(3D,Kpt,Mview,Img)	105
17.5	Body(3D,Kpt,Sview,Img)	106
17.6	Body(3D,Kpt,Sview,Vid)	106
17.7	Body(3D,Mesh,Sview,Img)	106
17.8	Face	107
17.9	Fashion	108
17.10	Hand(2D,Kpt,Rgb,Img)	108
17.11	Hand(3D,Kpt,Rgb,Img)	110
17.12	Hand(Gesture,Rgb,Vid)	110
17.13	Wholebody	110
18	Animal	113
18.1	Animalpose Dataset	113
18.2	Ap10k Dataset	114
18.3	Atrw Dataset	115
18.4	Fly Dataset	116
18.5	Horse10 Dataset	116
18.6	Locust Dataset	117
18.7	Macaque Dataset	118
18.8	Zebra Dataset	119
19	Body(2D,Kpt,Sview,Img)	121
19.1	Aic Dataset	121
19.2	Coco Dataset	123
19.3	Crowdpose Dataset	146
19.4	H36m Dataset	149
19.5	JHMDB Dataset	149
19.6	MHP Dataset	150
19.7	Mpii Dataset	152
19.8	Mpii_trb Dataset	160
19.9	Ochuman Dataset	160
19.10	Posetrack18 Dataset	162
20	Body(2D,Kpt,Sview,Vid)	165
20.1	Posetrack18 Dataset	165

21	Body(3D,Kpt,Sview,Img)	167
21.1	H36m Dataset	167
21.2	Mpi_inf_3dhp Dataset	168
22	Body(3D,Kpt,Sview,Vid)	169
22.1	H36m Dataset	169
22.2	Mpi_inf_3dhp Dataset	170
23	Body(3D,Kpt,Mview,Img)	171
23.1	Campus Dataset	171
23.2	Panoptic Dataset	171
23.3	Shelf Dataset	172
24	Body(3D,Mesh,Sview,Img)	173
24.1	Mixed Dataset	173
25	Face	175
25.1	300w Dataset	175
25.2	Aflw Dataset	175
25.3	Coco_wholebody_face Dataset	177
25.4	Cofw Dataset	180
25.5	WFLW Dataset	180
26	Fashion	185
26.1	Deepfashion Dataset	185
26.2	Deepfashion2 Dataset	187
27	Hand(2D,Kpt,Rgb,Img)	189
27.1	Coco_wholebody_hand Dataset	189
27.2	Freihand2d Dataset	193
27.3	Interhand2d Dataset	193
27.4	Onehand10k Dataset	194
27.5	Panoptic2d Dataset	198
27.6	Rhd2d Dataset	201
28	Hand(3D,Kpt,Rgb,Img)	205
28.1	Interhand3d Dataset	205
29	Hand(Gesture,Rgb,Vid)	207
29.1	Nvgesture Dataset	207
30	Wholebody	209
30.1	Coco-Wholebody Dataset	209
30.2	Halpe Dataset	213
31	Algorithms	215
31.1	MTUT (CVPR'2019)	215
31.2	MSPN (ArXiv'2019)	216
31.3	InterNet (ECCV'2020)	216
31.4	DEKR (CVPR'2021)	217
31.5	HigherHRNet (CVPR'2020)	218
31.6	DeepPose (CVPR'2014)	222
31.7	RLE (ICCV'2021)	230
31.8	SoftWingloss (TIP'2021)	232
31.9	VideoPose3D (CVPR'2019)	233
31.10	Hourglass (ECCV'2016)	234

31.11	LiteHRNet (CVPR'2021)	236
31.12	AdaptiveWingloss (ICCV'2019)	238
31.13	SimpleBaseline2D (ECCV'2018)	238
31.14	PoseWarper (NeurIPS'2019)	257
31.15	SimpleBaseline3D (ICCV'2017)	258
31.16	HMR (CVPR'2018)	259
31.17	UDP (CVPR'2020)	260
31.18	ViPNAS (CVPR'2021)	264
31.19	Wingloss (CVPR'2018)	266
31.20	DarkPose (CVPR'2020)	267
31.21	Associative Embedding (NIPS'2017)	276
31.22	VoxelPose (ECCV'2020)	285
31.23	RSN (ECCV'2020)	286
31.24	CID (CVPR'2022)	287
31.25	CPM (CVPR'2016)	287
31.26	HRNet (CVPR'2019)	289
31.27	HRNetv2 (TPAMI'2019)	306
31.28	SCNet (CVPR'2020)	317
32	Backbones	321
32.1	MSPN (ArXiv'2019)	321
32.2	MobilenetV2 (CVPR'2018)	321
32.3	HigherHRNet (CVPR'2020)	326
32.4	ResNeSt (ArXiv'2020)	329
32.5	ResNext (CVPR'2017)	330
32.6	ResNet (CVPR'2016)	331
32.7	Hourglass (ECCV'2016)	354
32.8	ShufflenetV1 (CVPR'2018)	356
32.9	Swin (ICCV'2021)	357
32.10	LiteHRNet (CVPR'2021)	358
32.11	I3D (CVPR'2017)	360
32.12	ShufflenetV2 (ECCV'2018)	360
32.13	TCFormer (CVPR'2022)	361
32.14	ResNetV1D (CVPR'2019)	362
32.15	SEResNet (CVPR'2018)	363
32.16	ViPNAS (CVPR'2021)	364
32.17	PVTv2 (CVMJ'2022)	365
32.18	PVT (ICCV'2021)	366
32.19	RSN (ECCV'2020)	367
32.20	CPM (CVPR'2016)	368
32.21	HRNet (CVPR'2019)	369
32.22	HRNetv2 (TPAMI'2019)	386
32.23	HRFormer (NIPS'2021)	397
32.24	AlexNet (NeurIPS'2012)	398
32.25	VGG (ICLR'2015)	398
32.26	SCNet (CVPR'2020)	399
33	Datasets	403
33.1	InterHand2.6M (ECCV'2020)	403
33.2	MPII (CVPR'2014)	404
33.3	NVGesture (CVPR'2016)	412
33.4	MacaquePose (bioRxiv'2020)	413
33.5	ATRW (ACM MM'2020)	414
33.6	Animal-Pose (ICCV'2019)	415

33.7	Human3.6M (TPAMI'2014)	416
33.8	OneHand10K (TCSVT'2019)	418
33.9	JHMDB (ICCV'2013)	422
33.10	CMU Panoptic (ICCV'2015)	423
33.11	CMU Panoptic HandDB (CVPR'2017)	424
33.12	AFLW (ICCVW'2011)	427
33.13	300W (IMAVIS'2016)	429
33.14	FreiHand (ICCV'2019)	429
33.15	MHP (ACM MM'2018)	430
33.16	Vinegar Fly (Nature Methods'2019)	431
33.17	CrowdPose (CVPR'2019)	432
33.18	Halpe (CVPR'2020)	434
33.19	OCHuman (CVPR'2019)	435
33.20	AP-10K (NeurIPS'2021)	436
33.21	Desert Locust (Elife'2019)	437
33.22	COCO-WholeBody-Face (ECCV'2020)	438
33.23	MPI-INF-3DHP (3DV'2017)	441
33.24	AI Challenger (ArXiv'2017)	442
33.25	MPII-TRB (ICCV'2019)	445
33.26	WFLW (CVPR'2018)	445
33.27	DeepFashion2 (CVPR'2019)	450
33.28	PoseTrack18 (CVPR'2018)	450
33.29	COCO-WholeBody-Hand (ECCV'2020)	453
33.30	COFW (ICCV'2013)	456
33.31	DeepFashion (CVPR'2016)	457
33.32	COCO-WholeBody (ECCV'2020)	458
33.33	Horse-10 (WACV'2021)	463
33.34	Grévy's Zebra (Elife'2019)	464
33.35	COCO (ECCV'2014)	464
33.36	RHD (ICCV'2017)	488
34	Techniques	493
34.1	RLE (ICCV'2021)	493
34.2	SoftWingloss (TIP'2021)	495
34.3	FP16 (ArXiv'2017)	496
34.4	Albumentations (Information'2020)	497
34.5	AdaptiveWingloss (ICCV'2019)	498
34.6	FPN (CVPR'2017)	499
34.7	UDP (CVPR'2020)	500
34.8	Wingloss (CVPR'2018)	504
34.9	DarkPose (CVPR'2020)	505
35	Tutorial 0: Learn about Configs	515
35.1	Modify config through script arguments	515
35.2	Config File Naming Convention	516
35.3	FAQ	520
36	Tutorial 1: Finetuning Models	521
36.1	Outline	521
36.2	Modify Head	521
36.3	Modify dataset	523
36.4	Modify training schedule	523
36.5	Use pre-trained model	524
37	Tutorial 2: Adding New Dataset	525

37.1	Customize datasets by reorganizing data to COCO format	525
37.2	Create a custom dataset_info config file for the dataset	526
37.3	Create a custom dataset class	530
37.4	Create a custom training config file	530
38	Tutorial 3: Custom Data Pipelines	533
38.1	Design of Data pipelines	533
38.2	Extend and use custom pipelines	535
39	Tutorial 4: Adding New Modules	537
39.1	Customize optimizer	537
39.2	Customize optimizer constructor	538
40	Tutorial 5: Exporting a model to ONNX	541
40.1	Supported Models	541
40.2	Usage	541
41	Tutorial 6: Customize Runtime Settings	543
41.1	Customize Optimization Methods	544
41.2	Customize Training Schedules	546
41.3	Customize Workflow	547
41.4	Customize Hooks	547
42	Tutorial 7: Develop Applications with Webcam API	551
42.1	Overview	551
42.2	An Example of Webcam Applications	552
42.3	Extending Webcam API with Custom Nodes	554
43	Useful Tools	559
43.1	Log Analysis	559
43.2	Model Complexity (Experimental)	561
43.3	Model Conversion	561
43.4	Model Serving	562
43.5	Miscellaneous	564
44	Changelog	565
44.1	v0.29.0 (14/10/2022)	565
44.2	v0.28.1 (28/07/2022)	566
44.3	v0.28.0 (06/07/2022)	566
44.4	v0.27.0 (07/06/2022)	566
44.5	v0.26.0 (05/05/2022)	567
44.6	v0.25.0 (02/04/2022)	569
44.7	v0.24.0 (07/03/2022)	570
44.8	v0.23.0 (11/02/2022)	570
44.9	v0.22.0 (04/01/2022)	571
44.10	v0.21.0 (06/12/2021)	572
44.11	v0.20.0 (01/11/2021)	573
44.12	v0.19.0 (08/10/2021)	573
44.13	v0.18.0 (01/09/2021)	574
44.14	v0.17.0 (06/08/2021)	575
44.15	v0.16.0 (02/07/2021)	576
44.16	v0.15.0 (02/06/2021)	576
44.17	v0.14.0 (06/05/2021)	577
44.18	v0.13.0 (31/03/2021)	578
44.19	v0.12.0 (28/02/2021)	578

44.20	v0.11.0 (31/01/2021)	579
44.21	v0.10.0 (31/12/2020)	580
44.22	v0.9.0 (30/11/2020)	581
44.23	v0.8.0 (31/10/2020)	582
44.24	v0.7.0 (30/9/2020)	583
44.25	v0.6.0 (31/8/2020)	584
44.26	v0.5.0 (21/7/2020)	585
45	FAQ	587
45.1	Installation	587
45.2	Coding	588
45.3	Data	588
45.4	Training	589
45.5	Evaluation	589
45.6	Inference	590
45.7	Deployment	590
46	mmpose.apis	591
47	mmpose.apis.webcam	599
47.1	Executor	599
47.2	Nodes	601
47.3	Utils	618
48	mmpose.core	627
48.1	evaluation	627
48.2	fp16	636
48.3	utils	638
48.4	post_processing	639
49	mmpose.models	645
49.1	backbones	645
49.2	necks	675
49.3	detectors	679
49.4	heads	707
49.5	losses	726
49.6	misc	738
50	mmpose.datasets	739
50.1	datasets	790
50.2	pipelines	808
50.3	samplers	820
51	mmpose.utils	821
52	English	823
53		825
54	Indices and tables	827
	Python Module Index	829
	Index	831

You can change the documentation language at the lower-left corner of the page.

- *Prerequisites*
- *Installation*
 - *Best Practices*
 - *Verify the installation*
 - *Customize Installation*
 - * *CUDA versions*
 - * *Install MMCV without MIM*
 - * *Install on CPU-only platforms*
 - * *Install on Google Colab*
 - * *Using MMPose with Docker*
 - *Trouble shooting*

PREREQUISITES

In this section we demonstrate how to prepare an environment with PyTorch.

MMPose works on Linux, Windows and macOS. It requires Python 3.6+, CUDA 9.2+ and PyTorch 1.5+.

If you are experienced with PyTorch and have already installed it, just skip this part and jump to the [next section](#). Otherwise, you can follow these steps for the preparation.

Step 0. Download and install Miniconda from the [official website](#).

Step 1. Create a conda environment and activate it.

```
conda create --name openmmlab python=3.8 -y
conda activate openmmlab
```

Step 2. Install PyTorch following [official instructions](#), e.g.

On GPU platforms:

```
conda install pytorch torchvision -c pytorch
```

Warning: This command will automatically install the latest version PyTorch and cudatoolkit, please check whether they match your environment.

On CPU platforms:

```
conda install pytorch torchvision cpuonly -c pytorch
```


INSTALLATION

We recommend that users follow our best practices to install MMPose. However, the whole process is highly customizable. See [Customize Installation](#) section for more information.

2.1 Best Practices

Step 0. Install [MMCV](#) using [MIM](#).

```
pip install -U openmim
mim install mmcv-full
```

Step 1. Install MMPose.

Case a: If you develop and run mmpose directly, install it from source:

```
git clone https://github.com/open-mmlab/mmpose.git
cd mmpose
pip install -r requirements.txt
pip install -v -e .
# "-v" means verbose, or more output
# "-e" means installing a project in editable mode,
# thus any local modifications made to the code will take effect without reinstallation.
```

Case b: If you use mmpose as a dependency or third-party package, install it with pip:

```
pip install mmpose
```

2.2 Verify the installation

To verify that MMPose is installed correctly, you can run an inference demo with the following steps.

Step 1. We need to download config and checkpoint files.

```
mim download mmpose --config associative_embedding_hrnet_w32_coco_512x512 --dest .
```

The downloading will take several seconds or more, depending on your network environment. When it is done, you will find two files `associative_embedding_hrnet_w32_coco_512x512.py` and `hrnet_w32_coco_512x512-bcb8c247_20200816.pth` in your current folder.

Step 2. Verify the inference demo.

Option (a). If you install mmpose from source, just run the following command.

```
python demo/bottom_up_img_demo.py associative_embedding_hrnet_w32_coco_512x512.py hrnet_
↪w32_coco_512x512-bcb8c247_20200816.pth --img-path tests/data/coco/ --out-img-root vis_
↪results
```

You will see several images in this folder: `vis_results`, where the human pose estimation results are plotted on the images.

Option (b). If you install mmpose with pip, open you python interpreter and copy&paste the following codes.

```
from mmpose.apis import (init_pose_model, inference_bottom_up_pose_model, vis_pose_
↪result)

config_file = 'associative_embedding_hrnet_w32_coco_512x512.py'
checkpoint_file = 'hrnet_w32_coco_512x512-bcb8c247_20200816.pth'
pose_model = init_pose_model(config_file, checkpoint_file, device='cpu') # or device=
↪'cuda:0'

image_name = 'demo/persons.jpg'
# test a single image
pose_results, _ = inference_bottom_up_pose_model(pose_model, image_name)

# show the results
vis_pose_result(pose_model, image_name, pose_results, out_file='demo/vis_persons.jpg')
```

Prepare an image with persons and place it properly, then run the above code, and you will see the output image with detected human poses plotted on it.

2.3 Customize Installation

2.3.1 CUDA versions

When installing PyTorch, you need to specify the version of CUDA. If you are not clear on which to choose, follow our recommendations:

- For Ampere-based NVIDIA GPUs, such as GeForce 30 series and NVIDIA A100, CUDA 11 is a must.
- For older NVIDIA GPUs, CUDA 11 is backward compatible, but CUDA 10.2 offers better compatibility and is more lightweight.

Please make sure the GPU driver satisfies the minimum version requirements. See [this table](#) for more information.

Installing CUDA runtime libraries is enough if you follow our best practices, because no CUDA code will be compiled locally. However if you hope to compile MMCV from source or develop other CUDA operators, you need to install the complete CUDA toolkit from NVIDIA's [website](#), and its version should match the CUDA version of PyTorch. i.e., the specified version of cudatoolkit in conda `install` command.

2.3.2 Install MMCV without MIM

MMCV contains C++ and CUDA extensions, thus depending on PyTorch in a complex way. MIM solves such dependencies automatically and makes the installation easier. However, it is not a must.

To install MMCV with pip instead of MIM, please follow [MMCV installation guides](#). This requires manually specifying a find-url based on PyTorch version and its CUDA version.

For example, the following command install mmcv-full built for PyTorch 1.10.x and CUDA 11.3.

```
pip install mmcv-full -f https://download.openmmlab.com/mmcv/dist/cu113/torch1.10/index.html
```

2.3.3 Install on CPU-only platforms

MMPose can be built for CPU only environment. In CPU mode you can train (requires MMCV version $\geq 1.4.4$), test or inference a model.

However, some functionalities are missing in this mode, usually GPU-compiled ops like **Deformable Convolution**. Most models in MMPose don't depend on these ops, but if you try to train/test/infer a model containing these ops, an error will be raised.

2.3.4 Install on Google Colab

[Google Colab](#) usually has PyTorch installed, thus we only need to install MMCV and MMPose with the following commands.

Step 1. Install MMCV using MIM.

```
!pip3 install openmim
!mim install mmcv-full
```

Step 2. Install MMPose from the source.

```
!git clone https://github.com/open-mmlab/mmpose.git
%cd mmpose
!pip install -e .
```

Step 3. Verification.

```
import mmpose
print(mmpose.__version__)
# Example output: 0.26.0
```

Note: Note that within Jupyter, the exclamation mark ! is used to call external executables and %cd is a [magic command](#) to change the current working directory of Python.

2.3.5 Using MMPose with Docker

We provide a [Dockerfile](#) to build an image. Ensure that your `docker` version ≥ 19.03 .

```
# build an image with PyTorch 1.6.0, CUDA 10.1, CUDNN 7.  
# If you prefer other versions, just modified the Dockerfile  
docker build -t mmpose docker/
```

Important: Make sure you've installed the [nvidia-container-toolkit](#).

Run it with

```
docker run --gpus all --shm-size=8g -it -v {DATA_DIR}:/mmpose/data mmpose
```

{DATA_DIR} is your local folder containing all the datasets for mmpose.

Note: If you encounter the error message like `permission denied`, please add `sudo` at the start of the command and try it again.

2.4 Trouble shooting

If you have some issues during the installation, please first view the [FAQ](#) page. You may [open an issue](#) on GitHub if no solution is found.

GET STARTED

This page provides basic tutorials about the usage of MMPose. For installation instructions, please see [install.md](#).

- *Prepare Datasets*
- *Inference with Pre-Trained Models*
 - *Test a dataset*
 - *Run demos*
- *Train a Model*
 - *Training setting*
 - *Train with a single GPU*
 - *Train with CPU*
 - *Train with multiple GPUs*
 - *Train with multiple machines*
 - *Launch multiple jobs on a single machine*
- *Benchmark*
- *Tutorials*

3.1 Prepare Datasets

MMPose supports multiple tasks. Please follow the corresponding guidelines for data preparation.

- *2D Body Keypoint Detection*
- *3D Body Keypoint Detection*
- *3D Body Mesh Recovery*
- *2D Hand Keypoint Detection*
- *3D Hand Keypoint Detection*
- *2D Face Keypoint Detection*
- *2D WholeBody Keypoint Detection*
- *2D Fashion Landmark Detection*
- *2D Animal Keypoint Detection*

3.2 Inference with Pre-trained Models

We provide testing scripts to evaluate a whole dataset (COCO, MAPII etc.), and provide some high-level apis for easier integration to other OpenMMLab projects.

3.2.1 Test a dataset

- [x] single GPU
- [x] CPU
- [x] single node multiple GPUs
- [x] multiple node

You can use the following commands to test a dataset.

```
# single-gpu testing
python tools/test.py ${CONFIG_FILE} ${CHECKPOINT_FILE} [--out ${RESULT_FILE}] [--fuse-conv-bn] \
    [--eval ${EVAL_METRICS}] [--gpu_collect] [--tmpdir ${TMPDIR}] [--cfg-options ${CFG_OPTIONS}] \
    [--launcher ${JOB_LAUNCHER}] [--local_rank ${LOCAL_RANK}]

# CPU: disable GPUs and run single-gpu testing script
export CUDA_VISIBLE_DEVICES=-1
python tools/test.py ${CONFIG_FILE} ${CHECKPOINT_FILE} [--out ${RESULT_FILE}] \
    [--eval ${EVAL_METRICS}]

# multi-gpu testing
./tools/dist_test.sh ${CONFIG_FILE} ${CHECKPOINT_FILE} ${GPU_NUM} [--out ${RESULT_FILE}] \
    [--fuse-conv-bn] \
    [--eval ${EVAL_METRIC}] [--gpu_collect] [--tmpdir ${TMPDIR}] [--cfg-options ${CFG_OPTIONS}] \
    [--launcher ${JOB_LAUNCHER}] [--local_rank ${LOCAL_RANK}]
```

Note that the provided CHECKPOINT_FILE is either the path to the model checkpoint file downloaded in advance, or the url link to the model checkpoint.

Optional arguments:

- RESULT_FILE: Filename of the output results. If not specified, the results will not be saved to a file.
- --fuse-conv-bn: Whether to fuse conv and bn, this will slightly increase the inference speed.
- EVAL_METRICS: Items to be evaluated on the results. Allowed values depend on the dataset.
- --gpu_collect: If specified, recognition results will be collected using gpu communication. Otherwise, it will save the results on different gpus to TMPDIR and collect them by the rank 0 worker.
- TMPDIR: Temporary directory used for collecting results from multiple workers, available when --gpu_collect is not specified.
- CFG_OPTIONS: Override some settings in the used config, the key-value pair in xxx=yyy format will be merged into config file. For example, 'cfg-options model.backbone.depth=18 model.backbone.with_cp=True'.
- JOB_LAUNCHER: Items for distributed job initialization launcher. Allowed choices are none, pytorch, slurm, mpi. Especially, if set to none, it will test in a non-distributed mode.

- LOCAL_RANK: ID for local rank. If not specified, it will be set to 0.

Examples:

Assume that you have already downloaded the checkpoints to the directory `checkpoints/`.

1. Test ResNet50 on COCO (without saving the test results) and evaluate the mAP.

```
./tools/dist_test.sh configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_
↪coco_256x192.py \
    checkpoints/SOME_CHECKPOINT.pth 1 \
    --eval mAP
```

2. Test ResNet50 on COCO with 8 GPUS. Download the checkpoint via url, and evaluate the mAP.

```
./tools/dist_test.sh configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_
↪coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_coco_256x192-
↪ec54d7f3_20200709.pth 8 \
    --eval mAP
```

3. Test ResNet50 on COCO in slurm environment and evaluate the mAP.

```
./tools/slurm_test.sh slurm_partition test_job \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_coco_256x192.py \
    checkpoints/SOME_CHECKPOINT.pth \
    --eval mAP
```

3.2.2 Run demos

We also provide scripts to run demos. Here is an example of running top-down human pose demos using ground-truth bounding boxes.

```
python demo/top_down_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
↪20200708.pth \
    --img-root tests/data/coco/ --json-file tests/data/coco/test_coco.json \
    --out-img-root vis_results
```

More examples and details can be found in the [demo folder](#) and the [demo docs](#).

3.3 Train a model

MMPose implements distributed training and non-distributed training, which uses `MMDistributedDataParallel` and `MMDDataParallel` respectively.

We adopt distributed training for both single machine and multiple machines. Supposing that the server has 8 GPUs, 8 processes will be started and each process runs on a single GPU.

Each process keeps an isolated model, data loader, and optimizer. Model parameters are only synchronized once at the beginning. After a forward and backward pass, gradients will be allreduced among all GPUs, and the optimizer will update model parameters. Since the gradients are allreduced, the model parameter stays the same for all processes after the iteration.

3.3.1 Training setting

All outputs (log files and checkpoints) will be saved to the working directory, which is specified by `work_dir` in the config file.

By default we evaluate the model on the validation set after each epoch, you can change the evaluation interval by modifying the interval argument in the training config

```
evaluation = dict(interval=5) # This evaluate the model per 5 epoch.
```

According to the [Linear Scaling Rule](#), you need to set the learning rate proportional to the batch size if you use different GPUs or videos per GPU, e.g., `lr=0.01` for 4 GPUs x 2 video/gpu and `lr=0.08` for 16 GPUs x 4 video/gpu.

3.3.2 Train with a single GPU

```
python tools/train.py ${CONFIG_FILE} [optional arguments]
```

If you want to specify the working directory in the command, you can add an argument `--work-dir ${YOUR_WORK_DIR}`.

3.3.3 Train with CPU

The process of training on the CPU is consistent with single GPU training. We just need to disable GPUs before the training process.

```
export CUDA_VISIBLE_DEVICES=-1
```

And then run the script above.

Note:

We do not recommend users to use CPU for training because it is too slow. We support this feature to allow users to debug on machines without GPU for convenience.

3.3.4 Train with multiple GPUs

```
./tools/dist_train.sh ${CONFIG_FILE} ${GPU_NUM} [optional arguments]
```

Optional arguments are:

- `--work-dir` `${WORK_DIR}`: Override the working directory specified in the config file.
- `--resume-from` `${CHECKPOINT_FILE}`: Resume from a previous checkpoint file.
- `--no-validate`: Whether not to evaluate the checkpoint during training.
- `--gpus` `${GPU_NUM}`: Number of gpus to use, which is only applicable to non-distributed training.
- `--gpu-ids` `${GPU_IDS}`: IDs of gpus to use, which is only applicable to non-distributed training.
- `--seed` `${SEED}`: Seed id for random state in python, numpy and pytorch to generate random numbers.
- `--deterministic`: If specified, it will set deterministic options for CUDNN backend.
- `--cfg-options` `CFG_OPTIONS`: Override some settings in the used config, the key-value pair in `xxx=yyy` format will be merged into config file. For example, `'--cfg-options model.backbone.depth=18 model.backbone.with_cp=True'`.
- `--launcher` `${JOB_LAUNCHER}`: Items for distributed job initialization launcher. Allowed choices are `none`, `pytorch`, `slurm`, `mpi`. Especially, if set to `none`, it will test in a non-distributed mode.
- `--autoscale-lr`: If specified, it will automatically scale lr with the number of gpus by [Linear Scaling Rule](#).
- `LOCAL_RANK`: ID for local rank. If not specified, it will be set to 0.

Difference between `resume-from` and `load-from`: `resume-from` loads both the model weights and optimizer status, and the epoch is also inherited from the specified checkpoint. It is usually used for resuming the training process that is interrupted accidentally. `load-from` only loads the model weights and the training epoch starts from 0. It is usually used for finetuning.

Here is an example of using 8 GPUs to load ResNet50 checkpoint.

```
./tools/dist_train.sh configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_coco_
↳ 256x192.py 8 --resume_from work_dirs/res50_coco_256x192/latest.pth
```

3.3.5 Train with multiple machines

If you can run MMPose on a cluster managed with [slurm](#), you can use the script `slurm_train.sh`. (This script also supports single machine training.)

```
./tools/slurm_train.sh ${PARTITION} ${JOB_NAME} ${CONFIG_FILE} ${WORK_DIR}
```

Here is an example of using 16 GPUs to train ResNet50 on the dev partition in a slurm cluster. (Use `GPUS_PER_NODE=8` to specify a single slurm cluster node with 8 GPUs, `CPUS_PER_TASK=2` to use 2 cpus per task. Assume that `Test` is a valid `${PARTITION}` name.)

```
GPUS=16 GPUS_PER_NODE=8 CPUS_PER_TASK=2 ./tools/slurm_train.sh Test res50 configs/body/
↳ 2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_coco_256x192.py work_dirs/res50_coco_
↳ 256x192
```

You can check [slurm_train.sh](#) for full arguments and environment variables.

If you launch with multiple machines simply connected with ethernet, you can simply run following commands:

```
# On the first machine:
NNODES=2 NODE_RANK=0 PORT=$MASTER_PORT MASTER_ADDR=$MASTER_ADDR ./tools/dist_train.sh
↪ $CONFIG $GPUS

# On the second machine:
NNODES=2 NODE_RANK=1 PORT=$MASTER_PORT MASTER_ADDR=$MASTER_ADDR ./tools/dist_train.sh
↪ $CONFIG $GPUS
```

Usually it is slow if you do not have high speed networking like InfiniBand.

3.3.6 Launch multiple jobs on a single machine

If you launch multiple jobs on a single machine, e.g., 2 jobs of 4-GPU training on a machine with 8 GPUs, you need to specify different ports (29500 by default) for each job to avoid communication conflict.

If you use `dist_train.sh` to launch training jobs, you can set the port in commands.

```
CUDA_VISIBLE_DEVICES=0,1,2,3 PORT=29500 ./tools/dist_train.sh ${CONFIG_FILE} 4
CUDA_VISIBLE_DEVICES=4,5,6,7 PORT=29501 ./tools/dist_train.sh ${CONFIG_FILE} 4
```

If you use launch training jobs with slurm, you need to modify the config files (usually the 4th line in config files) to set different communication ports.

In `config1.py`,

```
dist_params = dict(backend='nccl', port=29500)
```

In `config2.py`,

```
dist_params = dict(backend='nccl', port=29501)
```

Then you can launch two jobs with `config1.py` and `config2.py`.

```
CUDA_VISIBLE_DEVICES=0,1,2,3 ./tools/slurm_train.sh ${PARTITION} ${JOB_NAME} config1.py $
↪ ${WORK_DIR} 4
CUDA_VISIBLE_DEVICES=4,5,6,7 ./tools/slurm_train.sh ${PARTITION} ${JOB_NAME} config2.py $
↪ ${WORK_DIR} 4
```

3.4 Benchmark

You can get average inference speed using the following script. Note that it does not include the IO time and the pre-processing time.

```
python tools/analysis/benchmark_inference.py ${MMPOSE_CONFIG_FILE}
```


3.5 Tutorials

We provide some tutorials for users:

- *learn about configs*
- *finetune model*
- *add new dataset*
- *customize data pipelines*
- *add new modules*
- *export a model to ONNX*
- *customize runtime settings*
- *develop applications with Webcam API*

4.1 2D Animal Pose Demo

4.1.1 2D Animal Pose Image Demo

Using gt bounding boxes as input

We provide a demo script to test a single image, given gt json file.

Pose Model Preparation: The pre-trained pose estimation model can be downloaded from [model zoo](#). Take [macaque model](#) as an example:

```
python demo/top_down_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo.py \
    configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap/macaque/res50_macaque_256x192.py \
    ↪ https://download.openmmlab.com/mmpose/animal/resnet/res50_macaque_256x192-98f1dd3a_
    ↪ 20210407.pth \
    --img-root tests/data/macaque/ --json-file tests/data/macaque/test_macaque.json \
    --out-img-root vis_results
```

To run demos on CPU:

```
python demo/top_down_img_demo.py \
    configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap/macaque/res50_macaque_256x192.py \
    ↪ https://download.openmmlab.com/mmpose/animal/resnet/res50_macaque_256x192-98f1dd3a_
    ↪ 20210407.pth \
    --img-root tests/data/macaque/ --json-file tests/data/macaque/test_macaque.json \
    --out-img-root vis_results \
    --device=cpu
```

4.1.2 2D Animal Pose Video Demo

We also provide video demos to illustrate the results.

Using the full image as input

If the video is cropped with the object centered in the screen, we can simply use the full image as the model input (without object detection).

```
python demo/top_down_video_demo_full_frame_without_det.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/top_down_video_demo_full_frame_without_det.py \
    configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap/fly/res152_fly_192x192.py \
    https://download.openmmlab.com/mmpose/animal/resnet/res152_fly_192x192-fcafb5a_
↪ 20210407.pth \
    --video-path https://user-images.githubusercontent.com/87690686/165095600-f68e0d42-
↪ 830d-4c22-8940-c90c9f3bb817.mp4 \
    --out-video-root vis_results
```

Using MMDetection to detect animals

Assume that you have already installed `mmdet`.

COCO-animals

In COCO dataset, there are 80 object categories, including 10 common animal categories (15: 'bird', 16: 'cat', 17: 'dog', 18: 'horse', 19: 'sheep', 20: 'cow', 21: 'elephant', 22: 'bear', 23: 'zebra', 24: 'giraffe') For these COCO-animals, please download the COCO pre-trained detection model from [MMDetection Model Zoo](#).

```
python demo/top_down_video_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    --det-cat-id ${CATEGORY_ID} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/top_down_video_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_2x_
↪coco/faster_rcnn_r50_fpn_2x_coco_bbox_mAP-0.384_20200504_210434-a5d8aa15.pth \
    configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap/horse10/res50_horse10_256x256-
↪split1.py \
    https://download.openmmlab.com/mmpose/animal/resnet/res50_horse10_256x256_split1-
↪3a3dc37e_20210405.pth \
    --video-path https://user-images.githubusercontent.com/15977946/173124855-c626835e-
↪1863-4003-8184-315bc0b7b561.mp4 \
    --out-video-root vis_results \
    --bbox-thr 0.1 \
    --kpt-thr 0.4 \
    --det-cat-id 18
```

Other Animals

For other animals, we have also provided some pre-trained animal detection models (1-class models). Supported models can be found in [det model zoo](#). The pre-trained animal pose estimation model can be found in [pose model zoo](#).

```
python demo/top_down_video_demo_with_mmdet.py \
    ${MMDNET_CONFIG_FILE} ${MMDNET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--det-cat-id ${CATEGORY_ID}]
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/top_down_video_demo_with_mmdet.py \
    demo/mmdetection_cfg/cascade_rcnn_x101_64x4d_fpn_1class.py \
    https://download.openmmlab.com/mmpose/mmdet_pretrained/cascade_rcnn_x101_64x4d_fpn_
↪20e_macaque-e45e36f5_20210409.pth \
    configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap/macaque/hrnet_w32_macaque_
↪256x192.py \
    https://download.openmmlab.com/mmpose/animal/hrnet/hrnet_w32_macaque_256x192-
↪f7e9e04f_20210407.pth \
    --video-path https://user-images.githubusercontent.com/15977946/173135633-1c54a944-
↪4f01-4747-8c2e-55b8c83be533.mp4 \
    --out-video-root vis_results \
    --bbox-thr 0.5 \
    --kpt-thr 0.3 \
    --radius 9 \
    --thickness 3
```

4.1.3 Speed Up Inference

Some tips to speed up MMPose inference:

For 2D animal pose estimation models, try to edit the config file. For example,

1. set `flip_test=False` in `macaque-res50`.
2. set `post_process='default'` in `macaque-res50`.

4.2 2D Face Keypoint Demo

4.2.1 2D Face Image Demo

Using gt face bounding boxes as input

We provide a demo script to test a single image, given gt json file.

Face Keypoint Model Preparation: The pre-trained face keypoint estimation model can be found from [model zoo](#). Take [aflw model](#) as an example:

```
python demo/top_down_img_demo.py \
    ${MMPPOSE_CONFIG_FILE} ${MMPPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo.py \
    configs/face/2d_kpt_sview_rgb_img/topdown_heatmap/aflw/hrnetv2_w18_aflw_256x256.py \
    https://download.openmmlab.com/mmpose/face/hrnetv2/hrnetv2_w18_aflw_256x256-f2bbc62b_
↪20210125.pth \
    --img-root tests/data/aflw/ --json-file tests/data/aflw/test_aflw.json \
    --out-img-root vis_results
```

To run demos on CPU:

```
python demo/top_down_img_demo.py \
    configs/face/2d_kpt_sview_rgb_img/topdown_heatmap/aflw/hrnetv2_w18_aflw_256x256.py \
    https://download.openmmlab.com/mmpose/face/hrnetv2/hrnetv2_w18_aflw_256x256-f2bbc62b_
↪20210125.pth \
    --img-root tests/data/aflw/ --json-file tests/data/aflw/test_aflw.json \
    --out-img-root vis_results \
    --device=cpu
```

Using face bounding box detectors

We provide a demo script to run face detection and face keypoint estimation.

Please install `face_recognition` before running the demo, by `pip install face_recognition`. For more details, please refer to https://github.com/ageitgey/face_recognition.

```
python demo/face_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --img ${IMG_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

```
python demo/face_img_demo.py \
    configs/face/2d_kpt_sview_rgb_img/topdown_heatmap/aflw/hrnetv2_w18_aflw_256x256.py \
    https://download.openmmlab.com/mmpose/face/hrnetv2/hrnetv2_w18_aflw_256x256-f2bbc62b_
    ↪20210125.pth \
    --img-root tests/data/aflw/ \
    --img image04476.jpg \
    --out-img-root vis_results
```

4.2.2 2D Face Video Demo

We also provide a video demo to illustrate the results.

Please install `face_recognition` before running the demo, by `pip install face_recognition`. For more details, please refer to https://github.com/ageitgey/face_recognition.

```
python demo/face_video_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/face_video_demo.py \
    configs/face/2d_kpt_sview_rgb_img/topdown_heatmap/aflw/hrnetv2_w18_aflw_256x256.py \
    https://download.openmmlab.com/mmpose/face/hrnetv2/hrnetv2_w18_aflw_256x256-f2bbc62b_
    ↪20210125.pth \
    --video-path https://user-images.githubusercontent.com/87690686/137441355-ec4da09c-
    ↪3a8f-421b-bee9-b8b26f8c2dd0.mp4 \
    --out-video-root vis_results
```

4.2.3 Speed Up Inference

Some tips to speed up MMPose inference:

For 2D face keypoint estimation models, try to edit the config file. For example,

1. set `flip_test=False` in `face-hrnetv2_w18`.
2. set `post_process='default'` in `face-hrnetv2_w18`.

4.3 2D Hand Keypoint Demo

4.3.1 2D Hand Image Demo

Using gt hand bounding boxes as input

We provide a demo script to test a single image, given gt json file.

Hand Pose Model Preparation: The pre-trained hand pose estimation model can be downloaded from [model zoo](#). Take [onehand10k model](#) as an example:

```
python demo/top_down_img_demo.py \
    ${MMPose_CONFIG_FILE} ${MMPose_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo.py \
    configs/hand/2d_kpt_sview_rgb_img/topdown_heatmap/onehand10k/res50_onehand10k_
↪ 256x256.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_onehand10k_256x256-
↪ e67998f6_20200813.pth \
    --img-root tests/data/onehand10k/ --json-file tests/data/onehand10k/test_onehand10k.
↪ json \
    --out-img-root vis_results
```

To run demos on CPU:

```
python demo/top_down_img_demo.py \
    configs/hand/2d_kpt_sview_rgb_img/topdown_heatmap/onehand10k/res50_onehand10k_
↪ 256x256.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_onehand10k_256x256-
↪ e67998f6_20200813.pth \
    --img-root tests/data/onehand10k/ --json-file tests/data/onehand10k/test_onehand10k.
↪ json \
    --out-img-root vis_results \
    --device=cpu
```


Using mmdet for hand bounding box detection

We provide a demo script to run mmdet for hand detection, and mmpose for hand pose estimation.

Assume that you have already installed `mmdet`.

Hand Box Model Preparation: The pre-trained hand box estimation model can be found in [det model zoo](#).

Hand Pose Model Preparation: The pre-trained hand pose estimation model can be downloaded from [pose model zoo](#).

```
python demo/top_down_img_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --img ${IMG_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID} or CPU] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

```
python demo/top_down_img_demo_with_mmdet.py demo/mmdetection_cfg/cascade_rcnn_x101_64x4d_
↪ fpn_1class.py \
    https://download.openmmlab.com/mmpose/mmdet_pretrained/cascade_rcnn_x101_64x4d_fpn_
↪ 20e_onehand10k-dac19597_20201030.pth \
    configs/hand/2d_kpt_sview_rgb_img/topdown_heatmap/onehand10k/res50_onehand10k_
↪ 256x256.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_onehand10k_256x256-
↪ e67998f6_20200813.pth \
    --img-root tests/data/onehand10k/ \
    --img 9.jpg \
    --out-img-root vis_results
```

4.3.2 2D Hand Video Demo

We also provide a video demo to illustrate the results.

Assume that you have already installed `mmdet`.

Hand Box Model Preparation: The pre-trained hand box estimation model can be found in [det model zoo](#).

Hand Pose Model Preparation: The pre-trained hand pose estimation model can be found in [pose model zoo](#).

```
python demo/top_down_video_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID} or CPU] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/top_down_video_demo_with_mmdet.py demo/mmdetection_cfg/cascade_rcnn_x101_
↪ 64x4d_fpn_1class.py \
    https://download.openmmlab.com/mmpose/mmdet_pretrained/cascade_rcnn_x101_64x4d_fpn_
↪ 20e_onehand10k-dac19597_20201030.pth \
```

(continues on next page)

(continued from previous page)

```

    configs/hand/2d_kpt_sview_rgb_img/topdown_heatmap/onehand10k/res50_onehand10k_
↪ 256x256.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_onehand10k_256x256-
↪ e67998f6_20200813.pth \
    --video-path https://user-images.githubusercontent.com/87690686/137441388-3ea93d26-
↪ 5445-4184-829e-bf7011def9e4.mp4 \
    --out-video-root vis_results

```

4.3.3 Speed Up Inference

Some tips to speed up MMPose inference:

For 2D hand pose estimation models, try to edit the config file. For example,

1. set `flip_test=False` in `hand-res50`.
2. set `post_process='default'` in `hand-res50`.

4.4 2D Human Pose Demo

4.4.1 2D Human Pose Top-Down Image Demo

Using gt human bounding boxes as input

We provide a demo script to test a single image, given gt json file.

```

python demo/top_down_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]

```

Examples:

```

python demo/top_down_img_demo.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
↪ 20200708.pth \
    --img-root tests/data/coco/ --json-file tests/data/coco/test_coco.json \
    --out-img-root vis_results

```

To run demos on CPU:

```

python demo/top_down_img_demo.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
↪ 20200708.pth \

```

(continues on next page)

(continued from previous page)

```
--img-root tests/data/coco/ --json-file tests/data/coco/test_coco.json \
--out-img-root vis_results \
--device=cpu
```

Using mmdet for human bounding box detection

We provide a demo script to run mmdet for human detection, and mmpose for pose estimation.

Assume that you have already installed `mmdet`.

```
python demo/top_down_img_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --img ${IMG_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
    →coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
    →20200708.pth \
    --img-root tests/data/coco/ \
    --img 000000196141.jpg \
    --out-img-root vis_results
```

4.4.2 2D Human Pose Top-Down Video Demo

We also provide a video demo to illustrate the results.

Assume that you have already installed `mmdet`.

```
python demo/top_down_video_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}] \
    [--use-multi-frames] [--online]
```

Note that

1. `${VIDEO_PATH}` can be the local path or **URL** link to video file.
2. You can turn on the `[--use-multi-frames]` option to use multi frames for inference in the pose estimation stage.

3. If the [--online] option is set to **True**, future frame information can **not** be used when using multi frames for inference in the pose estimation stage.

Examples:

For single-frame inference that do not rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_video_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪ coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
↪ 20200708.pth \
    --video-path demo/resources/demo.mp4 \
    --out-video-root vis_results
```

For multi-frame inference that rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_video_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪ coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_vid/posewarper/posetrack18/hrnet_w48_posetrack18_
↪ 384x288_posewarper_stage2.py \
    https://download.openmmlab.com/mmpose/top_down/posewarper/hrnet_w48_posetrack18_
↪ 384x288_posewarper_stage2-4abf88db_20211130.pth \
    --video-path https://user-images.githubusercontent.com/87690686/137440639-fb08603d-
↪ 9a35-474e-b65f-46b5c06b68d6.mp4 \
    --out-video-root vis_results \
    --use-multi-frames --online
```

Using the full image as input

We also provide a video demo which does not require human bounding box detection. If the video is cropped with the human centered in the screen, we can simply use the full image as the model input.

```
python demo/top_down_video_demo_full_frame_without_det.py \
    ${MMPose_CONFIG_FILE} ${MMPose_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Note that \${VIDEO_PATH} can be the local path or **URL** link to video file.

Examples:

```
python demo/top_down_video_demo_full_frame_without_det.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/vipnas_res50_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/vipnas/vipnas_res50_coco_256x192-
↪ cc43b466_20210624.pth \
    --video-path https://user-images.githubusercontent.com/87690686/169808764-29e5678c-
↪ 6762-4f43-8666-c3e60f94338f.mp4 \
    --show
```

We also provide a GPU version which can accelerate inference and save CPU workload. Assume that you have already installed `ffmpegcv`. If the `--nvdecode` option is turned on, the video reader can support NVIDIA-VIDEO-DECODING for some qualified Nvidia GPUs, which can further accelerate the inference.

```
python demo/top_down_video_demo_full_frame_without_det_gpuaccel.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}] \
    [--nvdecode]
```

Examples:

```
python demo/top_down_video_demo_full_frame_without_det_gpuaccel.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/vipnas_res50_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/vipnas/vipnas_res50_coco_256x192-
    ↪cc43b466_20210624.pth \
    --video-path https://user-images.githubusercontent.com/87690686/169808764-29e5678c-
    ↪6762-4f43-8666-c3e60f94338f.mp4 \
    --out-video-root vis_results
```

4.4.3 2D Human Pose Bottom-Up Image Demo

We provide a demo script to test a single image.

```
python demo/bottom_up_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-path ${IMG_PATH} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR} --pose-nms-thr ${POSE_NMS_THR}]
```

Examples:

```
python demo/bottom_up_img_demo.py \
    configs/body/2d_kpt_sview_rgb_img/associative_embedding/coco/hrnet_w32_coco_512x512.
    ↪py \
    https://download.openmmlab.com/mmpose/bottom_up/hrnet_w32_coco_512x512-bcb8c247_
    ↪20200816.pth \
    --img-path tests/data/coco/ \
    --out-img-root vis_results
```

4.4.4 2D Human Pose Bottom-Up Video Demo

We also provide a video demo to illustrate the results.

```
python demo/bottom_up_video_demo.py \  
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \  
    --video-path ${VIDEO_PATH} \  
    --out-video-root ${OUTPUT_VIDEO_ROOT} \  
    [--show --device ${GPU_ID or CPU}] \  
    [--kpt-thr ${KPT_SCORE_THR} --pose-nms-thr ${POSE_NMS_THR}]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/bottom_up_video_demo.py \  
    configs/body/2d_kpt_sview_rgb_img/associative_embedding/coco/hrnet_w32_coco_512x512.  
→py \  
    https://download.openmmlab.com/mmpose/bottom_up/hrnet_w32_coco_512x512-bcb8c247_  
→20200816.pth \  
    --video-path demo/resources/demo.mp4 \  
    --out-video-root vis_results
```

4.4.5 Speed Up Inference

Some tips to speed up MMPose inference:

For top-down models, try to edit the config file. For example,

1. set `flip_test=False` in `topdown-res50`.
2. set `post_process='default'` in `topdown-res50`.
3. use faster human bounding box detector, see [MMDetection](#).

For bottom-up models, try to edit the config file. For example,

1. set `flip_test=False` in `AE-res50`.
2. set `adjust=False` in `AE-res50`.
3. set `refine=False` in `AE-res50`.
4. use smaller input image size in `AE-res50`.

4.5 2D Pose Tracking Demo

4.5.1 2D Top-Down Video Human Pose Tracking Demo

We provide a video demo to illustrate the pose tracking results.

Assume that you have already installed `mmdet`.

```
python demo/top_down_pose_tracking_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}] \
    [--use-oks-tracking --tracking-thr ${TRACKING_THR} --euro] \
    [--use-multi-frames] [--online]
```

Note that

1. `${VIDEO_PATH}` can be the local path or **URL** link to video file.
2. You can turn on the `[--use-multi-frames]` option to use multi frames for inference in the pose estimation stage.
3. If the `[--online]` option is set to **True**, future frame information can **not** be used when using multi frames for inference in the pose estimation stage.

Examples:

For single-frame inference that do not rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_pose_tracking_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
    ↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_coco_256x192-ec54d7f3_
    ↪20200709.pth \
    --video-path demo/resources/demo.mp4 \
    --out-video-root vis_results
```

For multi-frame inference that rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_pose_tracking_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
    ↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_vid/posewarper/posetrack18/hrnet_w48_posetrack18_
    ↪384x288_posewarper_stage2.py \
    https://download.openmmlab.com/mmpose/top_down/posewarper/hrnet_w48_posetrack18_
    ↪384x288_posewarper_stage2-4abf88db_20211130.pth \
    --video-path https://user-images.githubusercontent.com/87690686/137440639-fb08603d-
    ↪9a35-474e-b65f-46b5c06b68d6.mp4 \
    --out-video-root vis_results \
    --use-multi-frames --online
```

4.5.2 2D Top-Down Video Human Pose Tracking Demo with MMTracking

MMTracking is an open source video perception toolbox based on PyTorch for tracking related tasks. Here we show how to utilize MMTracking and MMPose to achieve human pose tracking.

Assume that you have already installed `mmtracking`.

```
python demo/top_down_video_demo_with_mmtracking.py \
    ${MMTRACKING_CONFIG_FILE} \
    ${MMPPOSE_CONFIG_FILE} ${MMPPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}] \
    [--use-multi-frames] [--online]
```

Note that

1. `${VIDEO_PATH}` can be the local path or **URL** link to video file.
2. You can turn on the `[--use-multi-frames]` option to use multi frames for inference in the pose estimation stage.
3. If the `[--online]` option is set to **True**, future frame information can **not** be used when using multi frames for inference in the pose estimation stage.

Examples:

For single-frame inference that do not rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_pose_tracking_demo_with_mmtracking.py \
    demo/mmtracking_cfg/tracktor_faster-rcnn_r50_fpn_4e_mot17-private.py \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/res50_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/resnet/res50_coco_256x192-ec54d7f3_
    ↪ 20200709.pth \
    --video-path demo/resources/demo.mp4 \
    --out-video-root vis_results
```

For multi-frame inference that rely on extra frames to get the final results of the current frame, try this:

```
python demo/top_down_pose_tracking_demo_with_mmtracking.py \
    demo/mmtracking_cfg/tracktor_faster-rcnn_r50_fpn_4e_mot17-private.py \
    configs/body/2d_kpt_sview_rgb_vid/posewarper/posetrack18/hrnet_w48_posetrack18_
    ↪ 384x288_posewarper_stage2.py \
    https://download.openmmlab.com/mmpose/top_down/posewarper/hrnet_w48_posetrack18_
    ↪ 384x288_posewarper_stage2-4abf88db_20211130.pth \
    --video-path demo/resources/demo.mp4 \
    --out-video-root vis_results \
    --use-multi-frames --online
```


4.5.3 2D Bottom-Up Video Human Pose Tracking Demo

We also provide a pose tracking demo with bottom-up pose estimation methods.

```
python demo/bottom_up_pose_tracking_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID} or CPU] \
    [--kpt-thr ${KPT_SCORE_THR} --pose-nms-thr ${POSE_NMS_THR}]
    [--use-oks-tracking --tracking-thr ${TRACKING_THR} --euro]
```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```
python demo/bottom_up_pose_tracking_demo.py \
    configs/body/2d_kpt_sview_rgb_img/associative_embedding/coco/hrnet_w32_coco_512x512.
    py \
    https://download.openmmlab.com/mmpose/bottom_up/hrnet_w32_coco_512x512-bcb8c247_
    20200816.pth \
    --video-path demo/resources/demo.mp4 \
    --out-video-root vis_results
```

4.5.4 Speed Up Inference

Some tips to speed up MMPose inference:

For top-down models, try to edit the config file. For example,

1. set `flip_test=False` in `topdown-res50`.
2. set `post_process='default'` in `topdown-res50`.
3. use faster human detector or human tracker, see `MMDetection` or `MMTracking`.

For bottom-up models, try to edit the config file. For example,

1. set `flip_test=False` in `AE-res50`.
2. set `adjust=False` in `AE-res50`.
3. set `refine=False` in `AE-res50`.
4. use smaller input image size in `AE-res50`.

4.6 2D Human Whole-Body Pose Demo

4.6.1 2D Human Whole-Body Pose Top-Down Image Demo

Using gt human bounding boxes as input

We provide a demo script to test a single image, given gt json file.

```
python demo/top_down_img_demo.py \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --json-file ${JSON_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo.py \
    configs/wholebody/2d_kpt_sview_rgb_img/topdown_heatmap/coco-wholebody/hrnet_w48_coco_
↪wholebody_384x288_dark_plus.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_wholebody_
↪384x288_dark-f5726563_20200918.pth \
    --img-root tests/data/coco/ --json-file tests/data/coco/test_coco.json \
    --out-img-root vis_results
```

To run demos on CPU:

```
python demo/top_down_img_demo.py \
    configs/wholebody/2d_kpt_sview_rgb_img/topdown_heatmap/coco-wholebody/hrnet_w48_coco_
↪wholebody_384x288_dark_plus.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_wholebody_
↪384x288_dark-f5726563_20200918.pth \
    --img-root tests/data/coco/ --json-file tests/data/coco/test_coco.json \
    --out-img-root vis_results \
    --device=cpu
```

Using mmdet for human bounding box detection

We provide a demo script to run mmdet for human detection, and mmpose for pose estimation.

Assume that you have already installed `mmdet`.

```
python demo/top_down_img_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --img-root ${IMG_ROOT} --img ${IMG_FILE} \
    --out-img-root ${OUTPUT_DIR} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]
```

Examples:

```
python demo/top_down_img_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
```

(continues on next page)

(continued from previous page)

```

    configs/wholebody/2d_kpt_sview_rgb_img/topdown_heatmap/coco-wholebody/hrnet_w48_coco_
↪wholebody_384x288_dark_plus.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_wholebody_
↪384x288_dark-f5726563_20200918.pth \
    --img-root tests/data/coco/ \
    --img 000000196141.jpg \
    --out-img-root vis_results

```

4.6.2 2D Human Whole-Body Pose Top-Down Video Demo

We also provide a video demo to illustrate the results.

Assume that you have already installed `mmdet`.

```

python demo/top_down_video_demo_with_mmdet.py \
    ${MMDET_CONFIG_FILE} ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \
    --video-path ${VIDEO_PATH} \
    --out-video-root ${OUTPUT_VIDEO_ROOT} \
    [--show --device ${GPU_ID or CPU}] \
    [--bbox-thr ${BBOX_SCORE_THR} --kpt-thr ${KPT_SCORE_THR}]

```

Note that `${VIDEO_PATH}` can be the local path or **URL** link to video file.

Examples:

```

python demo/top_down_video_demo_with_mmdet.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/wholebody/2d_kpt_sview_rgb_img/topdown_heatmap/coco-wholebody/hrnet_w48_coco_
↪wholebody_384x288_dark_plus.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_wholebody_
↪384x288_dark-f5726563_20200918.pth \
    --video-path https://user-images.githubusercontent.com/87690686/137440639-fb08603d-
↪9a35-474e-b65f-46b5c06b68d6.mp4 \
    --out-video-root vis_results

```

4.6.3 Speed Up Inference

Some tips to speed up MMPose inference:

For top-down models, try to edit the config file. For example,

1. set `flip_test=False` in `pose_hrnet_w48_dark+`.
2. set `post_process='default'` in `pose_hrnet_w48_dark+`.
3. use faster human bounding box detector, see [MMDetection](#).

4.7 3D Mesh Demo

4.7.1 3D Mesh Recovery Demo

We provide a demo script to recover human 3D mesh from a single image.

```
python demo/mesh_img_demo.py \  
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \  
    --json-file ${JSON_FILE} \  
    --img-root ${IMG_ROOT} \  
    [--show] \  
    [--device ${GPU_ID or CPU}] \  
    [--out-img-root ${OUTPUT_DIR}]
```

Example:

```
python demo/mesh_img_demo.py \  
    configs/body/3d_mesh_sview_rgb_img/hmr/mixed/res50_mixed_224x224.py \  
    https://download.openmmlab.com/mmpose/mesh/hmr/hmr_mesh_224x224-c21e8229_20201015.  
→pth \  
    --json-file tests/data/h36m/h36m_coco.json \  
    --img-root tests/data/h36m \  
    --out-img-root vis_results
```

4.8 3D Hand Demo

4.8.1 3D Hand Estimation Image Demo

Using gt hand bounding boxes as input

We provide a demo script to test a single image, given gt json file.

```
python demo/interhand3d_img_demo.py \  
    ${MMPOSE_CONFIG_FILE} ${MMPOSE_CHECKPOINT_FILE} \  
    --json-file ${JSON_FILE} \  
    --img-root ${IMG_ROOT} \  
    [--camera-param-file ${CAMERA_PARAM_FILE}] \  
    [--gt-joints-file ${GT_JOINTS_FILE}] \  
    [--show] \  
    [--device ${GPU_ID or CPU}] \  
    [--out-img-root ${OUTPUT_DIR}] \  
    [--rebase-keypoint-height] \  
    [--show-ground-truth]
```

Example with gt keypoints and camera parameters:

```
python demo/interhand3d_img_demo.py \
  configs/hand/3d_kpt_sview_rgb_img/internet/interhand3d/res50_interhand3d_all_256x256.
↪py \
  https://download.openmmlab.com/mmpose/hand3d/internet/res50_intehand3d_all_256x256-
↪b9c1cf4c_20210506.pth \
  --json-file tests/data/interhand2.6m/test_interhand2.6m_data.json \
  --img-root tests/data/interhand2.6m \
  --camera-param-file tests/data/interhand2.6m/test_interhand2.6m_camera.json \
  --gt-joints-file tests/data/interhand2.6m/test_interhand2.6m_joint_3d.json \
  --out-img-root vis_results \
  --rebase-keypoint-height \
  --show-ground-truth
```

Example without gt keypoints and camera parameters:

```
python demo/interhand3d_img_demo.py \
  configs/hand/3d_kpt_sview_rgb_img/internet/interhand3d/res50_interhand3d_all_256x256.
↪py \
  https://download.openmmlab.com/mmpose/hand3d/internet/res50_intehand3d_all_256x256-
↪b9c1cf4c_20210506.pth \
  --json-file tests/data/interhand2.6m/test_interhand2.6m_data.json \
  --img-root tests/data/interhand2.6m \
  --out-img-root vis_results \
  --rebase-keypoint-height
```

4.9 3D Human Pose Demo

4.9.1 3D Human Pose Two-stage Estimation Image Demo

Using ground truth 2D poses as the 1st stage (pose detection) result, and inference the 2nd stage (2D-to-3D lifting)

We provide a demo script to test on single images with a given ground-truth Json file.

```
python demo/body3d_two_stage_img_demo.py \
  ${MMPOSE_CONFIG_FILE_3D} \
  ${MMPOSE_CHECKPOINT_FILE_3D} \
  --json-file ${JSON_FILE} \
  --img-root ${IMG_ROOT} \
  --only-second-stage \
  [--show] \
  [--device ${GPU_ID or CPU}] \
  [--out-img-root ${OUTPUT_DIR}] \
  [--rebase-keypoint-height] \
  [--show-ground-truth]
```

Example:

```
python demo/body3d_two_stage_img_demo.py \
    configs/body/3d_kpt_sview_rgb_img/pose_lift/h36m/simplebaseline3d_h36m.py \
    https://download.openmmlab.com/mmpose/body3d/simple_baseline/simple3Dbaseline_h36m-
    f0ad73a4_20210419.pth \
    --json-file tests/data/h36m/h36m_coco.json \
    --img-root tests/data/h36m \
    --camera-param-file tests/data/h36m/cameras.pkl \
    --only-second-stage \
    --out-img-root vis_results \
    --rebase-keypoint-height \
    --show-ground-truth
```

4.9.2 3D Human Pose Two-stage Estimation Video Demo

Using `mmdet` for human bounding box detection and top-down model for the 1st stage (2D pose detection), and inference the 2nd stage (2D-to-3D lifting)

Assume that you have already installed `mmdet`.

```
python demo/body3d_two_stage_video_demo.py \
    ${MMDET_CONFIG_FILE} \
    ${MMDET_CHECKPOINT_FILE} \
    ${MMPOSE_CONFIG_FILE_2D} \
    ${MMPOSE_CHECKPOINT_FILE_2D} \
    ${MMPOSE_CONFIG_FILE_3D} \
    ${MMPOSE_CHECKPOINT_FILE_3D} \
    --video-path ${VIDEO_PATH} \
    [--rebase-keypoint-height] \
    [--norm-pose-2d] \
    [--num-poses-vis NUM_POSES_VIS] \
    [--show] \
    [--out-video-root ${OUT_VIDEO_ROOT}] \
    [--device ${GPU_ID or CPU}] \
    [--det-cat-id DET_CAT_ID] \
    [--bbox-thr BBOX_THR] \
    [--kpt-thr KPT_THR] \
    [--use-oks-tracking] \
    [--tracking-thr TRACKING_THR] \
    [--euro] \
    [--radius RADIUS] \
    [--thickness THICKNESS] \
    [--use-multi-frames] [--online]
```

Note that

1. `${VIDEO_PATH}` can be the local path or **URL** link to video file.
2. You can turn on the `[--use-multi-frames]` option to use multi frames for inference in the 2D pose detection stage.
3. If the `[--online]` option is set to **True**, future frame information can **not** be used when using multi frames for inference in the 2D pose detection stage.

Examples:

During 2D pose detection, for single-frame inference that do not rely on extra frames to get the final results of the current frame, try this:

```
python demo/body3d_two_stage_video_demo.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
    https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
↪20200708.pth \
    configs/body/3d_kpt_sview_rgb_vid/video_pose_lift/h36m/videopose3d_h36m_243frames_
↪fullconv_supervised_cpn_ft.py \
    https://download.openmmlab.com/mmpose/body3d/videopose/videopose_h36m_243frames_
↪fullconv_supervised_cpn_ft-88f5abbb_20210527.pth \
    --video-path https://user-images.githubusercontent.com/87690686/164970135-b14e424c-
↪765a-4180-9bc8-fa8d6abc5510.mp4 \
    --out-video-root vis_results \
    --rebase-keypoint-height
```

During 2D pose detection, for multi-frame inference that rely on extra frames to get the final results of the current frame, try this:

```
python demo/body3d_two_stage_video_demo.py \
    demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py \
    https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_
↪coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth \
    configs/body/2d_kpt_sview_rgb_vid/posewarper/posetrack18/hrnet_w48_posetrack18_
↪384x288_posewarper_stage2.py \
    https://download.openmmlab.com/mmpose/top_down/posewarper/hrnet_w48_posetrack18_
↪384x288_posewarper_stage2-4abf88db_20211130.pth \
    configs/body/3d_kpt_sview_rgb_vid/video_pose_lift/h36m/videopose3d_h36m_243frames_
↪fullconv_supervised_cpn_ft.py \
    https://download.openmmlab.com/mmpose/body3d/videopose/videopose_h36m_243frames_
↪fullconv_supervised_cpn_ft-88f5abbb_20210527.pth \
    --video-path https://user-images.githubusercontent.com/87690686/164970135-b14e424c-
↪765a-4180-9bc8-fa8d6abc5510.mp4 \
    --out-video-root vis_results \
    --rebase-keypoint-height \
    --use-multi-frames --online
```

4.10 3D Multiview Human Pose Demo

4.10.1 3D Multiview Human Pose Estimation Image Demo

VoxelPose

We provide a demo script to test on multiview images with given camera parameters. To run the demo:

```
python demo/body3d_multiview_detect_and_regress_img_demo.py \
    ${MMPOSE_CONFIG_FILE} \
    ${MMPOSE_CHECKPOINT_FILE} \
```

(continues on next page)

(continued from previous page)

```
--out-img-root ${OUT_IMG_ROOT} \
--camera-param-file ${CAMERA_FILE} \
[--img-root ${IMG_ROOT}] \
[--visualize-single-view ${VIS_SINGLE_IMG}] \
[--device ${GPU_ID or CPU}] \
[--out-img-root ${OUTPUT_DIR}]
```

Example:

```
python demo/body3d_multiview_detect_and_regress_img_demo.py \
    configs/body/3d_kpt_mview_rgb_img/voxelpose/panoptic/voxelpose_prn64x64x64_
↪cpn80x80x20_panoptic_cam5.py \
    https://download.openmmlab.com/mmpose/body3d/voxelpose/voxelpose_prn64x64x64_
↪cpn80x80x20_panoptic_cam5-545c150e_20211103.pth \
    --out-img-root vis_results \
    --camera-param-file tests/data/panoptic_body3d/demo/camera_parameters.json \
    --visualize-single-view
```

Data Preparation

Currently, we only support CMU Panoptic data format. Users can leave the argument `--img-root` unset to automatically download our [default demo data](#) (~6M). Users can also use custom data, which should be organized as follow:

```
|— ${IMG_ROOT}
|   |— camera_parameters.json
|   |— camera0
|       |— 0.jpg
|       |— ...
|   |— camera1
|   |— ...
```

The camera parameters should be a dictionary that include a key “cameras”. Under the key “cameras” should be a list of dictionaries containing the camera parameters. Each dictionary under the list should include a key “name”, the value of which is the directory name of images of a certain camera view.

```
{
  "cameras": [
    {"name": "camera0", ...},
    {"name": "camera1", ...},
    ...
  ]
}
```


4.11 Hand Gesture Recognition Demo

We provide a demo for gesture recognition with MMPose. This demo is built upon [MMPose Webcam API](#).

4.11.1 Get started

Launch the demo from the mmpose root directory:

```
python demo/webcam_demo.py --config demo/webcam_cfg/gesture_recognition.py
```

4.11.2 Hotkeys

Note that the demo will automatically save the output video into a file `gesture.mp4`.

4.11.3 Configurations

Detailed configurations can be found in the [config file](#). And more information about the gesture recognition model used in the demo can be found at the [model page](#).

4.12 Webcam Demo

We provide a webcam demo tool which integrates detection and 2D pose estimation for humans and animals. It can also apply fun effects like putting on sunglasses or enlarging the eyes, based on the pose estimation results.

4.12.1 Get started

Launch the demo from the mmpose root directory:

```
## Run webcam demo with GPU
python demo/webcam_demo.py

## Run webcam demo with CPU
python demo/webcam_demo.py --cpu
```

The command above will use the default config file `demo/webcam_cfg/pose_estimation.py`. You can also specify the config file in the command:

```
## Use the config "pose_tracking.py" for higher inference speed
python demo/webcam_demo.py --config demo/webcam_cfg/pose_tracking.py
```

4.12.2 Hotkeys

Note that the demo will automatically save the output video into a file `webcam_demo.mp4`.

4.12.3 Usage and configurations

Detailed configurations can be found in the config file.

- **Configure detection models** Users can choose detection models from the [MMDetection Model Zoo](#). Just set the `model_config` and `model_checkpoint` in the detector node accordingly, and the model will be automatically downloaded and loaded.

```
## 'DetectorNode':
## This node performs object detection from the frame image using an
## MMDetection model.
dict(
    type='DetectorNode',
    name='detector',
    model_config='demo/mmdetection_cfg/'
    'ssdlite_mobilenetv2_scratch_600e_coco.py',
    model_checkpoint='https://download.openmmlab.com'
    '/mmdetection/v2.0/ssd/'
    'ssdlite_mobilenetv2_scratch_600e_coco/ssdlite_mobilenetv2_'
    'scratch_600e_coco_20210629_110627-974d9307.pth',
    input_buffer='_input_',
    output_buffer='det_result')
```

- **Configure pose estimation models** In this demo we use two [top-down](#) pose estimation models for humans and animals respectively. Users can choose models from the [MMPose Model Zoo](#). To apply different pose models on different instance types, you can add multiple pose estimator nodes with `cls_names` set accordingly.

```
## 'TopDownPoseEstimatorNode':
## This node performs keypoint detection from the frame image using an
## MMPose top-down model. Detection results is needed.
dict(
    type='TopDownPoseEstimatorNode',
    name='human pose estimator',
    model_config='configs/wholebody/2d_kpt_sview_rgb_img/'
    'topdown_heatmap/coco-wholebody/'
    'vipnas_mbv3_coco_wholebody_256x192_dark.py',
    model_checkpoint='https://openmmlab-share.oss-cn-hangz'
    'hou.aliyuncs.com/mmpose/top_down/vipnas/vipnas_mbv3_co'
    'co_wholebody_256x192_dark-e2158108_20211205.pth',
    labels=['person'],
    input_buffer='det_result',
    output_buffer='human_pose'),
dict(
    type='TopDownPoseEstimatorNode',
    name='animal pose estimator',
    model_config='configs/animal/2d_kpt_sview_rgb_img/topdown_heatmap'
    '/animalpose/hrnet_w32_animalpose_256x256.py',
    model_checkpoint='https://download.openmmlab.com/mmpose/animal/'
    'hrnet/hrnet_w32_animalpose_256x256-1aa7f075_20210426.pth',
```

(continues on next page)

(continued from previous page)

```
labels=['cat', 'dog', 'horse', 'sheep', 'cow'],  
input_buffer='human_pose',  
output_buffer='animal_pose')
```

- **Run the demo on a local video file** You can use local video files as the demo input by set camera_id to the file path.
- **The computer doesn't have a camera?** A smart phone can serve as a webcam via apps like [Camo](#) or [DroidCam](#).
- **Test the camera and display** Run follow command for a quick test of video capturing and displaying.

```
python demo/webcam_demo.py --config demo/webcam_cfg/test_camera.py
```


BENCHMARK

We compare our results with some popular frameworks and official releases in terms of speed and accuracy.

5.1 Comparison Rules

Here we compare our MMPose repo with other pose estimation toolboxes in the same data and model settings.

To ensure the fairness of the comparison, the comparison experiments were conducted under the same hardware environment and using the same dataset. For each model setting, we kept the same data pre-processing methods to make sure the same feature input. In addition, we also used Memcached, a distributed memory-caching system, to load the data in all the compared toolboxes. This minimizes the IO time during benchmark.

The time we measured is the average training time for an iteration, including data processing and model training. The training speed is measure with s/iter. The lower, the better.

5.1.1 Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

We demonstrate the superiority of our MMPose framework in terms of speed and accuracy on the standard COCO keypoint detection benchmark. The mAP (the mean average precision) is used as the evaluation metric.

5.2 Hardware

- 8 NVIDIA Tesla V100 (32G) GPUs
- Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz

5.3 Software Environment

- Python 3.7
- PyTorch 1.4
- CUDA 10.1
- CUDNN 7.6.03
- NCCL 2.4.08

INFERENCE SPEED

We summarize the model complexity and inference speed of major models in MMPose, including FLOPs, parameter counts and inference speeds on both CPU and GPU devices with different batch sizes. We also compare the mAP of different models on COCO human keypoint dataset, showing the trade-off between model performance and model complexity.

6.1 Comparison Rules

To ensure the fairness of the comparison, the comparison experiments are conducted under the same hardware and software environment using the same dataset. We also list the mAP (mean average precision) on COCO human keypoint dataset of the models along with the corresponding config files.

For model complexity information measurement, we calculate the FLOPs and parameter counts of a model with corresponding input shape. Note that some layers or ops are currently not supported, for example, DeformConv2d, so you may need to check if all ops are supported and verify that the flops and parameter counts computation is correct.

For inference speed, we omit the time for data pre-processing and only measure the time for model forwarding and data post-processing. For each model setting, we keep the same data pre-processing methods to make sure the same feature input. We measure the inference speed on both CPU and GPU devices. For topdown heatmap models, we also test the case when the batch size is larger, e.g., 10, to test model performance in crowded scenes.

The inference speed is measured with frames per second (FPS), namely the average iterations per second, which can show how fast the model can handle an input. The higher, the faster, the better.

6.1.1 Hardware

- GPU: GeForce GTX 1660 SUPER
- CPU: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz

6.1.2 Software Environment

- Ubuntu 16.04
- Python 3.8
- PyTorch 1.10
- CUDA 10.2
- mmdcv-full 1.3.17
- mmpose 0.20.0

6.2 Model complexity information and inference speed results of major models in MMPose

1 Note that we run multiple iterations and record the time of each iteration, and the mean and standard deviation value of FPS are both shown.

2 The FPS is defined as the average iterations per second, regardless of the batch size in this iteration.

OVERVIEW

- Number of papers: 37
 - DATASET: 37

For supported pose algorithms, see [modelzoo overview](#).

7.1 2D Animal Keypoint Dataset

- Number of papers: 7
 - [DATASET] Ap-10k: A Benchmark for Animal Pose Estimation in the Wild (AP-10K)
 - [DATASET] Atrw: A Benchmark for Amur Tiger Re-Identification in the Wild (ATRW)
 - [DATASET] Cross-Domain Adaptation for Animal Pose Estimation (Animal-Pose)
 - [DATASET] Deepposekit, a Software Toolkit for Fast and Robust Animal Pose Estimation Using Deep Learning (Grévy’s Zebra , Desert Locust)
 - [DATASET] Fast Animal Pose Estimation Using Deep Neural Networks (Vinegar Fly)
 - [DATASET] Macaquepose: A Novel ‘In the Wild’ macaque Monkey Pose Dataset for Markerless Motion Capture (MacaquePose)
 - [DATASET] Pretraining Boosts Out-of-Domain Robustness for Pose Estimation (Horse-10)

7.2 2D Body Keypoint Datasets

- Number of papers: 9
 - [DATASET] 2d Human Pose Estimation: New Benchmark and State of the Art Analysis (MPII)
 - [DATASET] Ai Challenger: A Large-Scale Dataset for Going Deeper in Image Understanding (AIC)
 - [DATASET] Crowdpose: Efficient Crowded Scenes Pose Estimation and a New Benchmark (CrowdPose)
 - [DATASET] Learning Delicate Local Representations for Multi-Person Pose Estimation (sub-JHMDB dataset)
 - [DATASET] Microsoft Coco: Common Objects in Context (COCO)
 - [DATASET] Pose2seg: Detection Free Human Instance Segmentation (OCHuman)
 - [DATASET] Posetrack: A Benchmark for Human Pose Estimation and Tracking (PoseTrack18)
 - [DATASET] Trb: A Novel Triplet Representation for Understanding 2d Human Body (MPII-TRB)

- [DATASET] Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and a New Benchmark for Multi-Human Parsing (MHP)

7.3 2D Face Keypoint Datasets

- Number of papers: 5
 - [DATASET] 300 Faces in-the-Wild Challenge: Database and Results (300W Dataset)
 - [DATASET] Annotated Facial Landmarks in the Wild: A Large-Scale, Real-World Database for Facial Landmark Localization (AFLW Dataset)
 - [DATASET] Look at Boundary: A Boundary-Aware Face Alignment Algorithm (WFLW Dataset)
 - [DATASET] Robust Face Landmark Estimation Under Occlusion (COFW Dataset)
 - [DATASET] Whole-Body Human Pose Estimation in the Wild (COCO-WholeBody (Face))

7.4 2D Fashion Landmark Dataset

- Number of papers: 3
 - [DATASET] A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images (DeepFashion2)
 - [DATASET] Deepfashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations (DeepFashion (Fashion Landmark Detection, FLD))
 - [DATASET] Fashion Landmark Detection in the Wild (DeepFashion (Fashion Landmark Detection, FLD))

7.5 2D Hand Keypoint Datasets

- Number of papers: 1
 - [DATASET] Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3d Convolutional Neural Network (NVGesture)

7.6 2D Hand Keypoint Datasets

- Number of papers: 6
 - [DATASET] Freihand: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images (FreiHAND Dataset)
 - [DATASET] Hand Keypoint Detection in Single Images Using Multiview Bootstrapping (CMU Panoptic HandDB)
 - [DATASET] Interhand2.6m: A Dataset and Baseline for 3d Interacting Hand Pose Estimation From a Single RGB Image (InterHand2.6M)
 - [DATASET] Learning to Estimate 3d Hand Pose From Single RGB Images (RHD Dataset)

- [DATASET] Mask-Pose Cascaded CNN for 2d Hand Pose Estimation From Single Color Image (One-Hand10K)
- [DATASET] Whole-Body Human Pose Estimation in the Wild (COCO-WholeBody (Hand))

7.7 2D Wholebody Keypoint Datasets

- Number of papers: 2
 - [DATASET] Pastanet: Toward Human Activity Knowledge Engine (Halpe)
 - [DATASET] Whole-Body Human Pose Estimation in the Wild (COCO-WholeBody)

7.8 3D Body Keypoint Datasets

- Number of papers: 1
 - [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (Human3.6M)

7.9 3D Body Mesh Recovery Datasets

- Number of papers: 7
 - [DATASET] Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. (LSP)
 - [DATASET] End-to-End Recovery of Human Shape and Pose (CMU MoShed Data)
 - [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (Human3.6M)
 - [DATASET] Learning Effective Human Pose Estimation From Inaccurate Annotation (LSPET)
 - [DATASET] Microsoft Coco: Common Objects in Context (COCO)
 - [DATASET] Monocular 3d Human Pose Estimation in the Wild Using Improved CNN Supervision (MPI-INF-3DHP)
 - [DATASET] Smpl: A Skinned Multi-Person Linear Model (Notes)

7.10 3D Hand Keypoint Datasets

- Number of papers: 1
 - [DATASET] Interhand2.6m: A Dataset and Baseline for 3d Interacting Hand Pose Estimation From a Single RGB Image (InterHand2.6M)

2D BODY KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPOSE/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- Images
 - [COCO](#) [Homepage]
 - [MPII](#) [Homepage]
 - [MPII-TRB](#) [Homepage]
 - [AI Challenger](#) [Homepage]
 - [CrowdPose](#) [Homepage]
 - [OCHuman](#) [Homepage]
 - [MHP](#) [Homepage]
- Videos
 - [PoseTrack18](#) [Homepage]
 - [sub-JHMDB](#) [Homepage]

8.1 COCO

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

For [COCO](#) data, please download from [COCO download](#), 2017 Train/Val is needed for COCO keypoints training and validation. [HRNet-Human-Pose-Estimation](#) provides person detection result of COCO val2017 to reproduce our multi-person pose estimation results. Please download from [OneDrive](#) or [GoogleDrive](#). Optionally, to evaluate on COCO'2017 test-dev, please download the [image-info](#). Download and extract them under `$MMPOSE/data`, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── coco
    │   ├── annotations
    │   │   ├── person_keypoints_train2017.json
    │   │   ├── person_keypoints_val2017.json
    │   │   └── person_keypoints_test-dev-2017.json
    │   ├── person_detection_results
    │   │   ├── COCO_val2017_detections_AP_H_56_person.json
    │   │   └── COCO_test-dev2017_detections_AP_H_609_person.json
    │   ├── train2017
    │   │   ├── 00000000000009.jpg
    │   │   ├── 00000000000025.jpg
    │   │   ├── 00000000000030.jpg
    │   │   └── ...
    │   └── val2017
    │       ├── 00000000000139.jpg
    │       ├── 00000000000285.jpg
    │       ├── 00000000000632.jpg
    │       └── ...

```

8.2 MPII

```

@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

For [MPII](#) data, please download from [MPII Human Pose Dataset](#). We have converted the original annotation files into json format, please download them from [mpii_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── mpii
    │   └── annotations

```

(continues on next page)

(continued from previous page)

```

|   |— mpii_gt_val.mat
|   |— mpii_test.json
|   |— mpii_train.json
|   |— mpii_trainval.json
|   `— mpii_val.json
`— images
    |— 0000001163.jpg
    |— 0000003072.jpg

```

During training and inference, the prediction result will be saved as ‘.mat’ format by default. We also provide a tool to convert this ‘.mat’ to more readable ‘.json’ format.

```
python tools/dataset/mat2json ${PRED_MAT_FILE} ${GT_JSON_FILE} ${OUTPUT_PRED_JSON_FILE}
```

For example,

```
python tools/dataset/mat2json work_dirs/res50_mpii_256x256/pred.mat data/mpii/
↪ annotations/mpii_val.json pred.json
```

8.3 MPII-TRB

```

@inproceedings{duan2019trb,
  title={TRB: A Novel Triplet Representation for Understanding 2D Human Body},
  author={Duan, Haodong and Lin, Kwan-Yee and Jin, Sheng and Liu, Wentao and Qian, Chen_
↪ and Ouyang, Wanli},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={9479--9488},
  year={2019}
}

```

For MPII-TRB data, please download from [MPII Human Pose Dataset](#). Please download the annotation files from [mpii_trb_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```

mmpose
|— mmpose
|— docs
|— tests
|— tools
|— configs
`— data
    |— mpii
        |— annotations
        |   |— mpii_trb_train.json
        |   |— mpii_trb_val.json
        `— images
            |— 0000001163.jpg
            |— 0000003072.jpg

```

8.4 AIC

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
  ↪Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
  ↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

For AIC data, please download from [AI Challenger 2017](#), 2017 Train/Val is needed for keypoints training and validation. Please download the annotation files from [aic_annotations](#). Download and extract them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── aic
        ├── annotations
        │   ├── aic_train.json
        │   └── aic_val.json
        ├── ai_challenger_keypoint_train_20170902
        │   ├── keypoint_train_images_20170902
        │   │   ├── 0000252aea98840a550dac9a78c476ecb9f47ffa.jpg
        │   │   ├── 000050f770985ac9653198495ef9b5c82435d49c.jpg
        │   │   └── ...
        └── ai_challenger_keypoint_validation_20170911
            ├── keypoint_validation_images_20170911
            │   ├── 0002605c53fb92109a3f2de4fc3ce06425c3b61f.jpg
            │   ├── 0003b55a2c991223e6d8b4b820045bd49507bf6d.jpg
            │   └── ...
```

8.5 CrowdPose

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

For CrowdPose data, please download from [CrowdPose](#). Please download the annotation files and human detection results from [crowdpose_annotations](#). For top-down approaches, we follow CrowdPose to use the pre-trained weights of YOLOv3 to generate the detected human bounding boxes. For model training, we follow [HigherHRNet](#) to train models

on CrowdPose train/val dataset, and evaluate models on CrowdPose test dataset. Download and extract them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── crowdpose
        ├── annotations
        │   ├── mmpose_crowdpose_train.json
        │   ├── mmpose_crowdpose_val.json
        │   ├── mmpose_crowdpose_trainval.json
        │   ├── mmpose_crowdpose_test.json
        │   └── det_for_crowd_test_0.1_0.5.json
        └── images
            ├── 100000.jpg
            ├── 100001.jpg
            ├── 100002.jpg
            └── ...
```

8.6 OCHuman

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
  year={2019}
}
```

For OCHuman data, please download the images and annotations from OCHuman, Move them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── ochuman
        ├── annotations
        │   ├── ochuman_coco_format_val_range_0.00_1.00.json
        │   └── ochuman_coco_format_test_range_0.00_1.00.json
        └── images
            ├── 000001.jpg
```

(continues on next page)

(continued from previous page)

```
-- 000002.jpg
-- 000003.jpg
-- ...
```

8.7 MHP

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}
```

For [MHP](#) data, please download from [MHP](#). Please download the annotation files from [mhp_annotations](#). Please download and extract them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── mhp
    │   ├── annotations
    │   │   ├── mhp_train.json
    │   │   └── mhp_val.json
    │   ├── train
    │   │   ├── images
    │   │   │   ├── 1004.jpg
    │   │   │   ├── 10050.jpg
    │   │   │   └── ...
    │   ├── val
    │   │   ├── images
    │   │   │   ├── 10059.jpg
    │   │   │   ├── 10068.jpg
    │   │   │   └── ...
    │   └── test
    │       ├── images
    │       │   ├── 1005.jpg
    │       │   ├── 10052.jpg
    │       │   └── ...~~~~
    └── ...
```

8.8 PoseTrack18

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
  Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
  Recognition},
  pages={5167--5176},
  year={2018}
}
```

For PoseTrack18 data, please download from [PoseTrack18](#). Please download the annotation files from [pose-track18_annotations](#). We have merged the video-wise separated official annotation files into two json files (pose-track18_train & posetrack18_val.json). We also generate the [mask files](#) to speed up training. For top-down approaches, we use [MMDetection](#) pre-trained [Cascade R-CNN \(X-101-64x4d-FPN\)](#) to generate the detected human bounding boxes. Please download and extract them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── posetrack18
    │   ├── annotations
    │   │   ├── posetrack18_train.json
    │   │   ├── posetrack18_val.json
    │   │   └── posetrack18_val_human_detections.json
    │   ├── train
    │   │   ├── 000001_bonn_train.json
    │   │   ├── 000002_bonn_train.json
    │   │   └── ...
    │   ├── val
    │   │   ├── 000342_mpii_test.json
    │   │   ├── 000522_mpii_test.json
    │   │   └── ...
    │   └── test
    │       ├── 000001_mpiinew_test.json
    │       ├── 000002_mpiinew_test.json
    │       └── ...
    └── images
        ├── train
        │   ├── 000001_bonn_train
        │   │   ├── 000000.jpg
        │   │   ├── 000001.jpg
        │   │   └── ...
        │   └── ...
        └── val
            ├── 000342_mpii_test
            └── 000000.jpg
```

(continues on next page)

(continued from previous page)

```

    | | | | | -- 000001.jpg
    | | | | | -- ...
    | | | | | -- ...
    | | | | | -- test
    | | | | | -- 000001_mpiinew_test
    | | | | | -- 000000.jpg
    | | | | | -- 000001.jpg
    | | | | | -- ...
    | | | | | -- ...
    | | | | | -- mask
    | | | | | -- train
    | | | | | -- 000002_bonn_train
    | | | | | -- 000000.jpg
    | | | | | -- 000001.jpg
    | | | | | -- ...
    | | | | | -- ...
    | | | | | -- val
    | | | | | -- 000522_mpii_test
    | | | | | -- 000000.jpg
    | | | | | -- 000001.jpg
    | | | | | -- ...
    | | | | | -- ...

```

The official evaluation tool for PoseTrack should be installed from GitHub.

```
pip install git+https://github.com/svenkreiss/poseval.git
```

8.9 sub-JHMDB dataset

```

@misc{cai2020learning,
  title={Learning Delicate Local Representations for Multi-Person Pose Estimation},
  author={Yuanhao Cai and Zhicheng Wang and Zhengxiong Luo and Binyi Yin and Angang Du,
  ↪and Haoqian Wang and Xinyu Zhou and Erjin Zhou and Xiangyu Zhang and Jian Sun},
  year={2020},
  eprint={2003.04030},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```

For sub-JHMDB data, please download the images from JHMDB, Please download the annotation files from jhmdb_annotations. Move them under \$MMPOSE/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   └── jhmdb

```

(continues on next page)

(continued from previous page)

```
-- annotations
|  -- Sub1_train.json
|  -- Sub1_test.json
|  -- Sub2_train.json
|  -- Sub2_test.json
|  -- Sub3_train.json
|  -- Sub3_test.json
|-- Rename_Images
|  -- brush_hair
|  |  --April_09_brush_hair_u_nm_np1_ba_goo_0
|  |  |  --00001.png
|  |  |  --00002.png
|  -- catch
|  -- ...
```


2D WHOLEBODY KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPOSE/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *COCO-WholeBody* [[Homepage](#)]
- *Halpe* [[Homepage](#)]

9.1 COCO-WholeBody

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

For *COCO-WholeBody* dataset, images can be downloaded from [COCO download](#), 2017 Train/Val is needed for COCO keypoints training and validation. Download COCO-WholeBody annotations for COCO-WholeBody annotations for [Train / Validation](#) (Google Drive). Download person detection result of COCO val2017 from [OneDrive](#) or [GoogleDrive](#). Download and extract them under `$MMPOSE/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── coco
    │   ├── annotations
    │   │   ├── coco_wholebody_train_v1.0.json
    │   │   └── coco_wholebody_val_v1.0.json
    │   ├── person_detection_results
    │   │   └── COCO_val2017_detections_AP_H_56_person.json
    │   └── train2017
    │       ├── 00000000000009.jpg
    │       └── 00000000000025.jpg
```

(continues on next page)

(continued from previous page)

```

|   |   -- 0000000000030.jpg
|   |   -- ...
|   |   -- val2017
|   |   |   -- 0000000000139.jpg
|   |   |   -- 0000000000285.jpg
|   |   |   -- 0000000000632.jpg
|   |   |   -- ...

```

Please also install the latest version of [Extended COCO API](#) (version>=1.5) to support COCO-WholeBody evaluation:

```
pip install xtcocotools
```

9.2 Halpe

```

@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and
  Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},
  year={2020}
}

```

For [Halpe](#) dataset, please download images and annotations from [Halpe download](#). The images of the training set are from [HICO-Det](#) and those of the validation set are from [COCO](#). Download person detection result of COCO val2017 from [OneDrive](#) or [GoogleDrive](#). Download and extract them under \$MMPOSE/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   └── halpe
│       ├── annotations
│       │   ├── halpe_train_v1.json
│       │   └── halpe_val_v1.json
│       ├── person_detection_results
│       │   └── COCO_val2017_detections_AP_H_56_person.json
│       ├── hico_20160224_det
│       │   ├── anno_bbox.mat
│       │   ├── anno.mat
│       │   ├── README
│       │   ├── images
│       │   │   ├── train2015
│       │   │   │   ├── HICO_train2015_00000001.jpg
│       │   │   │   ├── HICO_train2015_00000002.jpg
│       │   │   │   ├── HICO_train2015_00000003.jpg
│       │   │   │   └── ...
│       │   └── test2015
│       └── tools

```

(continues on next page)

(continued from previous page)

```
| |-- ...  
|-- val2017  
| |-- 0000000000139.jpg  
| |-- 0000000000285.jpg  
| |-- 0000000000632.jpg  
| |-- ...
```

Please also install the latest version of [Extended COCO API](#) (version>=1.5) to support Halpe evaluation:

```
pip install xtcocotools
```


2D FACE KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPose/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *300W* [[Homepage](#)]
- *WFLW* [[Homepage](#)]
- *AFLW* [[Homepage](#)]
- *COFW* [[Homepage](#)]
- *COCO-WholeBody-Face* [[Homepage](#)]

10.1 300W Dataset

```
@article{sagonas2016300,
  title={300 faces in-the-wild challenge: Database and results},
  author={Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and
↪Zafeiriou, Stefanos and Pantic, Maja},
  journal={Image and vision computing},
  volume={47},
  pages={3--18},
  year={2016},
  publisher={Elsevier}
}
```

For 300W data, please download images from [300W Dataset](#). Please download the annotation files from [300w_annotations](#). Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── 300w
        ├── annotations
        │   ├── face_landmarks_300w_train.json
        │   └── face_landmarks_300w_valid.json
```

(continues on next page)

(continued from previous page)

```

|   |— face_landmarks_300w_valid_common.json
|   |— face_landmarks_300w_valid_challenge.json
|   |— face_landmarks_300w_test.json
|   `— images
|       |— afw
|           |— 1051618982_1.jpg
|           |— 111076519_1.jpg
|           ...
|       |— helen
|           |— trainset
|               |— 100032540_1.jpg
|               |— 100040721_1.jpg
|               ...
|           |— testset
|               |— 296814969_3.jpg
|               |— 2968560214_1.jpg
|               ...
|       |— ibug
|           |— image_003_1.jpg
|           |— image_004_1.jpg
|           ...
|       |— lfpw
|           |— trainset
|               |— image_0001.png
|               |— image_0002.png
|               ...
|           |— testset
|               |— image_0001.png
|               |— image_0002.png
|               ...
|   `— Test
|       |— 01_Indoor
|           |— indoor_001.png
|           |— indoor_002.png
|           ...
|       `— 02_Outdoor
|           |— outdoor_001.png
|           |— outdoor_002.png
|           ...

```

10.2 WFLW Dataset

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}

```

(continues on next page)

(continued from previous page)

}

For WFLW data, please download images from [WFLW Dataset](#). Please download the annotation files from [wflw_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   ├── wflw
│   │   ├── annotations
│   │   │   ├── face_landmarks_wflw_train.json
│   │   │   ├── face_landmarks_wflw_test.json
│   │   │   ├── face_landmarks_wflw_test_blur.json
│   │   │   ├── face_landmarks_wflw_test_occlusion.json
│   │   │   ├── face_landmarks_wflw_test_expression.json
│   │   │   ├── face_landmarks_wflw_test_largepose.json
│   │   │   ├── face_landmarks_wflw_test_illumination.json
│   │   │   └── face_landmarks_wflw_test_makeup.json
│   │   └── images
│   │       ├── 0--Parade
│   │       │   ├── 0_Parade_marchingband_1_1015.jpg
│   │       │   ├── 0_Parade_marchingband_1_1031.jpg
│   │       │   └── ...
│   │       ├── 1--Handshaking
│   │       │   ├── 1_Handshaking_Handshaking_1_105.jpg
│   │       │   ├── 1_Handshaking_Handshaking_1_107.jpg
│   │       │   └── ...
│   │       └── ...
│   └── ...
```

10.3 AFLW Dataset

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}
```

For AFLW data, please download images from [AFLW Dataset](#). Please download the annotation files from [aflw_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── aflw
    │   ├── annotations
    │   │   ├── face_landmarks_aflw_train.json
    │   │   ├── face_landmarks_aflw_test_frontal.json
    │   │   └── face_landmarks_aflw_test.json
    │   └── images
    │       ├── flickr
    │       │   ├── 0
    │       │   │   ├── image00002.jpg
    │       │   │   └── image00013.jpg
    │       │   └── ...
    │       │   ├── 2
    │       │   │   ├── image00004.jpg
    │       │   │   └── image00006.jpg
    │       │   └── ...
    │       └── 3
    │           ├── image00032.jpg
    │           └── image00035.jpg
    │           ...

```

10.4 COFW Dataset

```

@inproceedings{burgos2013robust,
  title={Robust face landmark estimation under occlusion},
  author={Burgos-Artiz, Xavier P and Perona, Pietro and Doll{\`a}, Piotr},
  booktitle={Proceedings of the IEEE international conference on computer vision},
  pages={1513--1520},
  year={2013}
}

```

For COFW data, please download from [COFW Dataset \(Color Images\)](#). Move COFW_train_color.mat and COFW_test_color.mat to data/cofw/ and make them look like:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── cofw
    │   ├── COFW_train_color.mat
    │   └── COFW_test_color.mat

```

Run the following script under {MMPose}/data

```
python tools/dataset/parse_cofw_dataset.py
```

And you will get

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── cofw
    │   ├── COFW_train_color.mat
    │   ├── COFW_test_color.mat
    │   ├── annotations
    │   │   ├── cofw_train.json
    │   │   └── cofw_test.json
    │   └── images
    │       ├── 000001.jpg
    │       └── 000002.jpg
```

10.5 COCO-WholeBody (Face)

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

For [COCO-WholeBody](#) dataset, images can be downloaded from [COCO download](#), 2017 Train/Val is needed for COCO keypoints training and validation. Download COCO-WholeBody annotations for COCO-WholeBody annotations for [Train / Validation](#) (Google Drive). Download person detection result of COCO val2017 from [OneDrive](#) or [GoogleDrive](#). Download and extract them under \$MMPOSE/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── coco
    │   ├── annotations
    │   │   ├── coco_wholebody_train_v1.0.json
    │   │   └── coco_wholebody_val_v1.0.json
    │   ├── person_detection_results
    │   │   ├── COCO_val2017_detections_AP_H_56_person.json
    │   └── train2017
```

(continues on next page)

(continued from previous page)

```
|      -- 00000000000009.jpg
|      -- 00000000000025.jpg
|      -- 00000000000030.jpg
|      -- ...
|-- val2017
|      -- 00000000000139.jpg
|      -- 00000000000285.jpg
|      -- 00000000000632.jpg
|      -- ...
```

Please also install the latest version of [Extended COCO API](#) to support COCO-WholeBody evaluation:

```
pip install xtcocotools
```


2D HAND KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPose/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *OneHand10K* [[Homepage](#)]
- *FreiHand* [[Homepage](#)]
- *CMU Panoptic HandDB* [[Homepage](#)]
- *InterHand2.6M* [[Homepage](#)]
- *RHD* [[Homepage](#)]
- *COCO-WholeBody-Hand* [[Homepage](#)]

11.1 OneHand10K

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

For *OneHand10K* data, please download from [OneHand10K Dataset](#). Please download the annotation files from [one-hand10k_annotations](#). Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── onehand10k
    │   └── annotations
```

(continues on next page)

(continued from previous page)

```

|   |— onehand10k_train.json
|   |— onehand10k_test.json
`— Train
|   |— source
|       |— 0.jpg
|       |— 1.jpg
|       ...
`— Test
    |— source
        |— 0.jpg
        |— 1.jpg

```

11.2 FreiHAND Dataset

```

@inproceedings{zimmermann2019freihand,
  title={Freihand: A dataset for markerless capture of hand pose and shape from single_
↪rgb images},
  author={Zimmermann, Christian and Ceylan, Duygu and Yang, Jimei and Russell, Bryan and_
↪Argus, Max and Brox, Thomas},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={813--822},
  year={2019}
}

```

For [FreiHAND](#) data, please download from [FreiHand Dataset](#). Since the official dataset does not provide validation set, we randomly split the training data into 8:1:1 for train/val/test. Please download the annotation files from [frei-hand_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── freihand
    │   ├── annotations
    │   │   ├── freihand_train.json
    │   │   ├── freihand_val.json
    │   │   └── freihand_test.json
    │   └── training
    │       ├── rgb
    │       │   ├── 00000000.jpg
    │       │   ├── 00000001.jpg
    │       │   └── ...
    │       ├── mask
    │       │   ├── 00000000.jpg
    │       │   ├── 00000001.jpg
    │       │   └── ...

```

11.3 CMU Panoptic HandDB

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

For [CMU Panoptic HandDB](#), please download from [CMU Panoptic HandDB](#). Following [Simon et al](#), panoptic images (hand143_panopticdb) and MPII & NZSL training sets (manual_train) are used for training, while MPII & NZSL test set (manual_test) for testing. Please download the annotation files from [panoptic_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── panoptic
    │   ├── annotations
    │   │   ├── panoptic_train.json
    │   │   └── panoptic_test.json
    │   └── hand143_panopticdb
    │       ├── imgs
    │       │   ├── 00000000.jpg
    │       │   ├── 00000001.jpg
    │       │   └── ...
    │       └── hand_labels
    │           ├── manual_train
    │           │   ├── 000015774_01_l.jpg
    │           │   ├── 000015774_01_r.jpg
    │           │   └── ...
    │           └── manual_test
    │               ├── 000648952_02_l.jpg
    │               ├── 000835470_01_l.jpg
    │               └── ...
```

11.4 InterHand2.6M

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation,
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

For [InterHand2.6M](#), please download from [InterHand2.6M](#). Please download the annotation files from [annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── interhand2.6m
    │   ├── annotations
    │   │   ├── all
    │   │   ├── human_annot
    │   │   ├── machine_annot
    │   │   ├── skeleton.txt
    │   │   └── subject.txt
    │   └── images
    │       ├── train
    │       │   ├── Capture0 ~ Capture26
    │       │   └── val
    │       ├── Capture0
    │       ├── test
    │       └── Capture0 ~ Capture7
```

11.5 RHD Dataset

```
@TechReport{zb2017hand,
author={Christian Zimmermann and Thomas Brox},
title={Learning to Estimate 3D Hand Pose from Single RGB Images},
institution={arXiv:1705.01389},
year={2017},
note="https://arxiv.org/abs/1705.01389",
url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

For [RHD Dataset](#), please download from [RHD Dataset](#). Please download the annotation files from [rhd_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── rhd
    │   ├── annotations
    │   │   ├── rhd_train.json
    │   │   └── rhd_test.json
    │   └── training
    │       ├── color
    │       │   ├── 00000.jpg
    │       │   └── 00001.jpg
    │       ├── depth
    │       │   ├── 00000.jpg
    │       │   └── 00001.jpg
    │       └── mask
    │           ├── 00000.jpg
    │           └── 00001.jpg
    └── evaluation
        ├── color
        │   ├── 00000.jpg
        │   └── 00001.jpg
        ├── depth
        │   ├── 00000.jpg
        │   └── 00001.jpg
        └── mask
            ├── 00000.jpg
            └── 00001.jpg

```

11.6 COCO-WholeBody (Hand)

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

For COCO-WholeBody dataset, images can be downloaded from [COCO download](#), 2017 Train/Val is needed for COCO keypoints training and validation. Download COCO-WholeBody annotations for COCO-WholeBody annotations for [Train / Validation](#) (Google Drive). Download person detection result of COCO val2017 from [OneDrive](#) or [GoogleDrive](#). Download and extract them under \$MMPOSE/data, and make them look like this:

```

mmpose
├── mmpose
├── docs

```

(continues on next page)

(continued from previous page)

```
├── tests
├── tools
├── configs
└── data
    ├── coco
    │   ├── annotations
    │   │   ├── coco_wholebody_train_v1.0.json
    │   │   └── coco_wholebody_val_v1.0.json
    │   ├── person_detection_results
    │   │   └── COCO_val2017_detections_AP_H_56_person.json
    │   ├── train2017
    │   │   ├── 00000000000009.jpg
    │   │   ├── 00000000000025.jpg
    │   │   ├── 00000000000030.jpg
    │   │   ├── ...
    │   └── val2017
    │       ├── 00000000000139.jpg
    │       ├── 00000000000285.jpg
    │       ├── 00000000000632.jpg
    │       └── ...
```

Please also install the latest version of [Extended COCO API](#) to support COCO-WholeBody evaluation:

```
pip install xtcocotools
```

2D FASHION LANDMARK DATASET

It is recommended to symlink the dataset root to `$MMPose/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- DeepFashion [[Homepage](#)]
- *DeepFashion2* [[Homepage](#)]

12.1 DeepFashion (Fashion Landmark Detection, FLD)

```
@inproceedings{liuLQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWTeccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

For DeepFashion dataset, images can be downloaded from [download](#). Please download the annotation files from `fld_annotations`. Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   └── fld
```

(continues on next page)

(continued from previous page)

```

-- annotations
|-- fld_upper_train.json
|-- fld_upper_val.json
|-- fld_upper_test.json
|-- fld_lower_train.json
|-- fld_lower_val.json
|-- fld_lower_test.json
|-- fld_full_train.json
|-- fld_full_val.json
|-- fld_full_test.json
-- img
|-- img_00000001.jpg
|-- img_00000002.jpg
|-- img_00000003.jpg
|-- img_00000004.jpg
|-- img_00000005.jpg
|-- ...

```

12.2 DeepFashion2

```

@article{DeepFashion2,
  author = {Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang,
↪and Ping Luo},
  title={A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-
↪Identification of Clothing Images},
  journal={CVPR},
  year={2019}
}

```

For DeepFashion2 dataset, images can be downloaded from [download](#). Please download the [annotation files](#). These annotation files are converted by `deepfashion2_to_coco.py`. Extract them under `{MMPose}/data`, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── deepfashion2
    │   ├── train
    │   │   ├── deepfashion2_short_sleeved_outwear_train.json
    │   │   ├── deepfashion2_short_sleeved_dress_train.json
    │   │   ├── deepfashion2_skirt_train.json
    │   │   ├── deepfashion2_sling_dress_train.json
    │   │   ├── ...
    │   │   └── image
    │   │       ├── 000001.jpg
    │   │       └── 000002.jpg

```

(continues on next page)

(continued from previous page)

```
|
|
| -- 000003.jpg
| -- 000004.jpg
| -- 000005.jpg
| -- ...
|-- validation
|   -- deepfashion2_short_sleeved_dress_validation.json
|   -- deepfashion2_long_sleeved_shirt_validation.json
|   -- deepfashion2_trousers_validation.json
|   -- deepfashion2_skirt_validation.json
|   -- ...
|   -- image
|     -- 000001.jpg
|     -- 000002.jpg
|     -- 000003.jpg
|     -- 000004.jpg
|     -- 000005.jpg
|     -- ...
```


2D ANIMAL KEYPOINT DATASET

It is recommended to symlink the dataset root to `$MMPose/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *Animal-Pose* [[Homepage](#)]
- *AP-10K* [[Homepage](#)]
- *Horse-10* [[Homepage](#)]
- *MacaquePose* [[Homepage](#)]
- *Vinegar Fly* [[Homepage](#)]
- *Desert Locust* [[Homepage](#)]
- Grévy's Zebra [[Homepage](#)]
- *ATRW* [[Homepage](#)]

13.1 Animal-Pose

```
@InProceedings{Cao_2019_ICCV,  
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,  
→ Cewu and Tai, Yu-Wing},  
  title = {Cross-Domain Adaptation for Animal Pose Estimation},  
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},  
  month = {October},  
  year = {2019}  
}
```

For *Animal-Pose* dataset, we prepare the dataset as follows:

1. Download the images of *PASCAL VOC2012*, especially the five categories (dog, cat, sheep, cow, horse), which we use as trainval dataset.
2. Download the *test-set* images with raw annotations (1000 images, 5 categories).
3. We have pre-processed the annotations to make it compatible with MMPose. Please download the annotation files from [annotations](#). If you would like to generate the annotations by yourself, please check our dataset parsing [codes](#).

Extract them under `{MMPose}/data`, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   └── animalpose
│       ├── VOC2012
│       │   ├── Annotations
│       │   ├── ImageSets
│       │   ├── JPEGImages
│       │   ├── SegmentationClass
│       │   └── SegmentationObject
│       ├── animalpose_image_part2
│       │   ├── cat
│       │   ├── cow
│       │   ├── dog
│       │   ├── horse
│       │   └── sheep
│       ├── annotations
│       │   ├── animalpose_train.json
│       │   ├── animalpose_val.json
│       │   ├── animalpose_trainval.json
│       │   └── animalpose_test.json
│       ├── PASCAL2011_animal_annotation
│       │   ├── cat
│       │   │   ├── 2007_000528_1.xml
│       │   │   ├── 2007_000549_1.xml
│       │   │   └── ...
│       │   ├── cow
│       │   ├── dog
│       │   ├── horse
│       │   └── sheep
│       ├── animalpose_anno2
│       │   ├── cat
│       │   │   ├── ca1.xml
│       │   │   ├── ca2.xml
│       │   │   └── ...
│       │   ├── cow
│       │   ├── dog
│       │   ├── horse
│       │   └── sheep

```

The official dataset does not provide the official train/val/test set split. We choose the images from PascalVOC for train & val. In total, we have 3608 images and 5117 annotations for train+val, where 2798 images with 4000 annotations are used for training, and 810 images with 1117 annotations are used for validation. Those images from other sources (1000 images with 1000 annotations) are used for testing.

13.2 AP-10K

```
@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng_
↪Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

For AP-10K dataset, images and annotations can be downloaded from [download](#). Note, this data and annotation data is for non-commercial use only.

Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   └── ap10k
│       ├── annotations
│       │   ├── ap10k-train-split1.json
│       │   ├── ap10k-train-split2.json
│       │   ├── ap10k-train-split3.json
│       │   ├── ap10k-val-split1.json
│       │   ├── ap10k-val-split2.json
│       │   ├── ap10k-val-split3.json
│       │   ├── ap10k-test-split1.json
│       │   ├── ap10k-test-split2.json
│       │   └── ap10k-test-split3.json
│       └── data
│           ├── 00000000000001.jpg
│           ├── 00000000000002.jpg
│           └── ...
```

The annotation files in ‘annotation’ folder contains 50 labeled animal species. There are total 10,015 labeled images with 13,028 instances in the AP-10K dataset. We randomly split them into train, val, and test set following the ratio of 7:1:2.

13.3 Horse-10

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekgonul, Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision},
  pages={1859--1868},
  year={2021}
}
```

For [Horse-10](#) dataset, images can be downloaded from [download](#). Please download the annotation files from [horse10_annotations](#). Note, this data and annotation data is for non-commercial use only, per the authors (see <http://horse10.deeplabcut.org> for more information). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── horse10
        ├── annotations
        │   ├── horse10-train-split1.json
        │   ├── horse10-train-split2.json
        │   ├── horse10-train-split3.json
        │   ├── horse10-test-split1.json
        │   ├── horse10-test-split2.json
        │   └── horse10-test-split3.json
        ├── labeled-data
        │   ├── BrownHorseinShadow
        │   ├── BrownHorseintoshadow
        │   └── ...
```

13.4 MacaquePose

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel ‘in the wild’ macaque monkey pose dataset for markerless motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru, Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

For [MacaquePose](#) dataset, images can be downloaded from [download](#). Please download the annotation files from

`macaque_annotations`. Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── macaque
        ├── annotations
        │   ├── macaque_train.json
        │   └── macaque_test.json
        └── images
            ├── 01418849d54b3005.jpg
            ├── 0142d1d1a6904a70.jpg
            ├── 01ef2c4c260321b7.jpg
            ├── 020a1c75c8c85238.jpg
            ├── 020b1506eef2557d.jpg
            └── ...
```

Since the official dataset does not provide the test set, we randomly select 12500 images for training, and the rest for evaluation (see `code`).

13.5 Vinegar Fly

```
@article{pereira2019fast,
  title={Fast animal pose estimation using deep neural networks},
  author={Pereira, Talmo D and Aldarondo, Diego E and Willmore, Lindsay and Kislin,
  ↪Mikhail and Wang, Samuel S-H and Murthy, Mala and Shaevitz, Joshua W},
  journal={Nature methods},
  volume={16},
  number={1},
  pages={117--125},
  year={2019},
  publisher={Nature Publishing Group}
}
```

For `Vinegar Fly` dataset, images can be downloaded from `vinegar_fly_images`. Please download the annotation files from `vinegar_fly_annotations`. Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── fly
        ├── annotations
        │   ├── fly_train.json
        │   └── fly_test.json
```

(continues on next page)

(continued from previous page)

```
-- images
  -- 0.jpg
  -- 1.jpg
  -- 2.jpg
  -- 3.jpg
  -- ...
```

Since the official dataset does not provide the test set, we randomly select 90% images for training, and the rest (10%) for evaluation (see [code](#)).

13.6 Desert Locust

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation_
↪ using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,_
↪ Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

For [Desert Locust](#) dataset, images can be downloaded from [locust_images](#). Please download the annotation files from [locust_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── locust
        ├── annotations
        │   ├── locust_train.json
        │   └── locust_test.json
        └── images
            ├── 0.jpg
            ├── 1.jpg
            ├── 2.jpg
            ├── 3.jpg
            └── ...
```

Since the official dataset does not provide the test set, we randomly select 90% images for training, and the rest (10%) for evaluation (see [code](#)).

13.7 Grévy's Zebra

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation_
↪using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,_
↪Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

For Grévy's Zebra dataset, images can be downloaded from [zebra_images](#). Please download the annotation files from [zebra_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── zebra
        ├── annotations
        │   ├── zebra_train.json
        │   └── zebra_test.json
        └── images
            ├── 0.jpg
            ├── 1.jpg
            ├── 2.jpg
            ├── 3.jpg
            └── ...
```

Since the official dataset does not provide the test set, we randomly select 90% images for training, and the rest (10%) for evaluation (see [code](#)).

13.8 ATRW

```
@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}
```

ATRW captures images of the Amur tiger (also known as Siberian tiger, Northeast-China tiger) in the wild. For ATRW dataset, please download images from [Pose_train](#), [Pose_val](#), and [Pose_test](#). Note that in the ATRW official annotation files, the key “file_name” is written as “filename”. To make it compatible with other coco-type json files, we

have modified this key. Please download the modified annotation files from [atrw_annotations](#). Extract them under {MMPose}/data, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── atrw
        ├── annotations
        │   ├── keypoint_train.json
        │   ├── keypoint_val.json
        │   └── keypoint_trainval.json
        ├── images
        │   ├── train
        │   │   ├── 000002.jpg
        │   │   ├── 000003.jpg
        │   │   └── ...
        │   ├── val
        │   │   ├── 000001.jpg
        │   │   ├── 000013.jpg
        │   │   └── ...
        │   └── test
        │       ├── 000000.jpg
        │       ├── 000004.jpg
        │       └── ...
```

3D BODY KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPPOSE/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *Human3.6M* [[Homepage](#)]
- *CMU Panoptic* [[Homepage](#)]
- *Campus/Shelf* [[Homepage](#)]

14.1 Human3.6M

```
@article{h36m_pami,  
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu,   
↪Cristian},  
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing   
↪in Natural Environments},  
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},  
  publisher = {IEEE Computer Society},  
  volume = {36},  
  number = {7},  
  pages = {1325-1339},  
  month = {jul},  
  year = {2014}  
}
```

For *Human3.6M*, please download from the official website and run the [preprocessing script](#), which will extract camera parameters and pose annotations at full framerate (50 FPS) and downsampled framerate (10 FPS). The processed data should have the following structure:

```
mmpose  
├── mmpose  
├── docs  
├── tests  
├── tools  
├── configs  
└── data  
    ├── h36m  
    │   ├── annotation_body3d  
    │   └── cameras.pkl
```

(continues on next page)

(continued from previous page)

```
|
|   |— fps50
|   |   |— h36m_test.npz
|   |   |— h36m_train.npz
|   |   |— joint2d_rel_stats.pkl
|   |   |— joint2d_stats.pkl
|   |   |— joint3d_rel_stats.pkl
|   |   `— joint3d_stats.pkl
|   `— fps10
|       |— h36m_test.npz
|       |— h36m_train.npz
|       |— joint2d_rel_stats.pkl
|       |— joint2d_stats.pkl
|       |— joint3d_rel_stats.pkl
|       `— joint3d_stats.pkl
`— images
    |— S1
    |   |— S1_Directions_1.54138969
    |   |   |— S1_Directions_1.54138969_00001.jpg
    |   |   |— S1_Directions_1.54138969_00002.jpg
    |   |   |— ...
    |   |— ...
    |— S5
    |— S6
    |— S7
    |— S8
    |— S9
    `— S11
```

Please note that Human3.6M dataset is also used in the 3D_body_mesh task, where different schemes for data preprocessing and organizing are adopted.

14.2 CMU Panoptic

```
@Article = {joo_iccv_2015,
author = {Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo
↳ Kanade, Shohei Nobuhara, and Yaser Sheikh},
title = {Panoptic Studio: A Massively Multiview System for Social Motion Capture},
booktitle = {ICCV},
year = {2015}
}
```

Please follow [voxelpose-pytorch](#) to prepare this dataset.

1. Download the dataset by following the instructions in [panoptic-toolbox](#) and extract them under `$MMPOSE/data/panoptic`.
2. Only download those sequences that are needed. You can also just download a subset of camera views by specifying the number of views (`HD_Video_Number`) and changing the camera order in `./scripts/getData.sh`. The used sequences and camera views can be found in [VoxelPose](#). Note that the sequence “160906_band3” might not be available due to errors on the server of CMU Panoptic.
3. Note that we only use HD videos, calibration data, and 3D Body Keypoint in the codes. You can comment out

other irrelevant codes such as downloading 3D Face data in `./scripts/getData.sh`.

The directory tree should be like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── panoptic
    │   ├── 16060224_hagglings1
    │   │   ├── hdimgs
    │   │   ├── hdvideos
    │   │   ├── hdPose3d_stage1_coco19
    │   │   └── calibration_160224_hagglings1.json
    │   └── 160226_hagglings1
    │       └── ...

```

14.3 Campus and Shelf

```

@inproceedings {belagian14multi,
  title = {{3D} Pictorial Structures for Multiple Human Pose Estimation},
  author = {Belagiannis, Vasileios and Amin, Sikandar and Andriluka, Mykhaylo and
↪Schiele, Bernt and Navab
  Nassir and Ilic, Slobodan},
  booktitle = {IEEE Computer Society Conference on Computer Vision and Pattern
↪Recognition (CVPR)},
  year = {2014},
  month = {June},
  organization={IEEE}
}

```

Please follow [voxelpose-pytorch](#) to prepare these two datasets.

1. Please download the datasets from the [official website](#) and extract them under `$MMPOSE/data/campus` and `$MMPOSE/data/shelf`, respectively. The original data include images as well as the ground truth pose file `actorsGT.mat`.
2. We directly use the processed camera parameters from [voxelpose-pytorch](#). You can download them from this repository and place in under `$MMPOSE/data/campus/calibration_campus.json` and `$MMPOSE/data/shelf/calibration_shelf.json`, respectively.
3. Like [Voxelpose](#), due to the limited and incomplete annotations of the two datasets, we don't train the model using this dataset. Instead, we directly use the 2D pose estimator trained on COCO, and use independent 3D human poses from the CMU Panoptic dataset to train our 3D model. It lies in ``${MMPOSE}/data/panoptic_training_pose.pkl`.
4. Like [Voxelpose](#), for testing, we first estimate 2D poses and generate 2D heatmaps for these two datasets. You can download the predicted poses from [voxelpose-pytorch](#) and place them in `$MMPOSE/data/campus/pred_campus_maskrcnn_hrnet_coco.pkl` and `$MMPOSE/data/shelf/pred_shelf_maskrcnn_hrnet_coco.pkl`, respectively. You can also use the models trained on COCO dataset (like HigherHRNet) to generate 2D heatmaps directly.

The directory tree should be like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── panoptic_training_pose.pkl
    ├── campus
    │   ├── Camera0
    │   │   ├── campus4-c0-00000.png
    │   │   ├── ...
    │   │   └── campus4-c0-01999.png
    │   ├── ...
    │   ├── Camera2
    │   │   ├── campus4-c2-00000.png
    │   │   ├── ...
    │   │   └── campus4-c2-01999.png
    │   ├── calibration_campus.json
    │   ├── pred_campus_maskrcnn_hrnet_coco.pkl
    │   └── actorsGT.mat
    ├── shelf
    │   ├── Camera0
    │   │   ├── img_000000.png
    │   │   ├── ...
    │   │   └── img_003199.png
    │   ├── ...
    │   ├── Camera4
    │   │   ├── img_000000.png
    │   │   ├── ...
    │   │   └── img_003199.png
    │   ├── calibration_shelf.json
    │   ├── pred_shelf_maskrcnn_hrnet_coco.pkl
    │   └── actorsGT.mat
```

3D BODY MESH RECOVERY DATASETS

It is recommended to symlink the dataset root to `$MMPOSE/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

To achieve high-quality human mesh estimation, we use multiple datasets for training. The following items should be prepared for human mesh training:

- *3D Body Mesh Recovery Datasets*
 - *Notes*
 - * *Annotation Files for Human Mesh Estimation*
 - * *SMPL Model*
 - *COCO*
 - *Human3.6M*
 - *MPI-INF-3DHP*
 - *LSP*
 - *LSPET*
 - *CMU MoShed Data*

15.1 Notes

15.1.1 Annotation Files for Human Mesh Estimation

For human mesh estimation, we use multiple datasets for training. The annotation of different datasets are preprocessed to the same format. Please follow the [preprocess procedure](#) of SPIN to generate the annotation files or download the processed files from [here](#), and make it look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── mesh_annotation_files
    │   ├── coco_2014_train.npz
    │   └── h36m_valid_protocol1.npz
```

(continues on next page)

(continued from previous page)

```

├── h36m_valid_protocol2.npz
├── hr-lspet_train.npz
├── lsp_dataset_original_train.npz
├── mpi_inf_3dhp_train.npz
└── mpii_train.npz

```

15.1.2 SMPL Model

```

@article{loper2015smpl,
  title={SMPL: A skinned multi-person linear model},
  author={Loper, Matthew and Mahmood, Naureen and Romero, Javier and Pons-Moll, Gerard_
↪and Black, Michael J},
  journal={ACM transactions on graphics (TOG)},
  volume={34},
  number={6},
  pages={1--16},
  year={2015},
  publisher={ACM New York, NY, USA}
}

```

For human mesh estimation, SMPL model is used to generate the human mesh. Please download the [gender neutral SMPL model](#), [joints regressor](#) and [mean parameters](#) under \$MMPOSE/models/smpl, and make it look like this:

```

mmpose
├── mmpose
├── ...
└── models
    └── smpl
        ├── joints_regressor_cmr.npy
        ├── smpl_mean_params.npz
        └── SMPL_NEUTRAL.pkl

```

15.2 COCO

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

For COCO data, please download from [COCO download](#). COCO'2014 Train is needed for human mesh estimation training. Download and extract them under \$MMPOSE/data, and make them look like this:


```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── coco
    │   ├── train2014
    │   │   ├── COCO_train2014_000000000009.jpg
    │   │   ├── COCO_train2014_000000000025.jpg
    │   │   ├── COCO_train2014_000000000030.jpg
    │   │   └── ...
    └── ...

```

15.3 Human3.6M

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}

```

For [Human3.6M](#), we use the MoShed data provided in [HMR](#) for training. However, due to license limitations, we are not allowed to redistribute the MoShed data.

For the evaluation on Human3.6M dataset, please follow the [preprocess procedure](#) of SPIN to extract test images from [Human3.6M](#) original videos, and make it look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── Human3.6M
    │   └── images
    │       ├── S11_Directions_1.54138969_000001.jpg
    │       ├── S11_Directions_1.54138969_000006.jpg
    │       ├── S11_Directions_1.54138969_000011.jpg
    │       └── ...
    └── ...

```

The download of Human3.6M dataset is quite difficult, you can also download the [zip file](#) of the test images. However, due to the license limitations, we are not allowed to redistribute the images either. So the users need to download the

original video and extract the images by themselves.

15.4 MPI-INF-3DHP

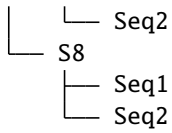
```
@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and
  ↪ Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
  ↪ ,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
  year = {2017},
  organization={IEEE},
  doi={10.1109/3dv.2017.00064},
}
```

For MPI-INF-3DHP, please follow the [preprocess procedure](#) of SPIN to sample images, and make them like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    └── mpi_inf_3dhp_test_set
        ├── TS1
        ├── TS2
        ├── TS3
        ├── TS4
        ├── TS5
        └── TS6
        ├── S1
        │   ├── Seq1
        │   └── Seq2
        ├── S2
        │   ├── Seq1
        │   └── Seq2
        ├── S3
        │   ├── Seq1
        │   └── Seq2
        ├── S4
        │   ├── Seq1
        │   └── Seq2
        ├── S5
        │   ├── Seq1
        │   └── Seq2
        ├── S6
        │   ├── Seq1
        │   └── Seq2
        └── S7
            └── Seq1
```

(continues on next page)

(continued from previous page)



15.5 LSP

```

@inproceedings{johnson2010clustered,
  title={Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation.},
  author={Johnson, Sam and Everingham, Mark},
  booktitle={bmvc},
  volume={2},
  number={4},
  pages={5},
  year={2010},
  organization={Citeseer}
}

```

For [LSP](#), please download the high resolution version [LSP dataset original](#). Extract them under `$MMPOSE/data`, and make them look like this:

```

mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
└── data
    ├── lsp_dataset_original
    │   └── images
    │       ├── im0001.jpg
    │       ├── im0002.jpg
    │       └── ...

```

15.6 LSPET

```

@inproceedings{johnson2011learning,
  title={Learning effective human pose estimation from inaccurate annotation},
  author={Johnson, Sam and Everingham, Mark},
  booktitle={CVPR 2011},
  pages={1465--1472},
  year={2011},
  organization={IEEE}
}

```

For [LSPET](#), please download its high resolution form [HR-LSPET](#). Extract them under `$MMPOSE/data`, and make them look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   ├── lspet_dataset
│   │   ├── images
│   │   │   ├── im000001.jpg
│   │   │   ├── im000002.jpg
│   │   │   ├── im000003.jpg
│   │   │   └── ...
│   └── joints.mat
```

15.7 CMU MoShed Data

```
@inproceedings{kanazawa2018end,
  title={End-to-end recovery of human shape and pose},
  author={Kanazawa, Angjoo and Black, Michael J and Jacobs, David W and Malik, Jitendra},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={7122--7131},
  year={2018}
}
```

Real-world SMPL parameters are used for the adversarial training in human mesh estimation. The MoShed data provided in [HMR](#) is included in this [zip file](#). Please download and extract it under `$MMPOSE/data`, and make it look like this:

```
mmpose
├── mmpose
├── docs
├── tests
├── tools
├── configs
├── data
│   ├── mesh_annotation_files
│   │   ├── CMU_mosh.npz
│   │   └── ...
```

3D HAND KEYPOINT DATASETS

It is recommended to symlink the dataset root to `$MMPose/data`. If your folder structure is different, you may need to change the corresponding paths in config files.

MMPose supported datasets:

- *InterHand2.6M* [[Homepage](#)]

16.1 InterHand2.6M

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,  
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,   
↪Kyoung Mu},  
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation   
↪from a Single RGB Image},  
booktitle = {European Conference on Computer Vision (ECCV)},  
year = {2020}  
}
```

For *InterHand2.6M*, please download from [InterHand2.6M](#). Please download the annotation files from [annotations](#). Extract them under `{MMPose}/data`, and make them look like this:

```
mmpose  
├── mmpose  
├── docs  
├── tests  
├── tools  
├── configs  
└── data  
    ├── interhand2.6m  
    │   ├── annotations  
    │   │   ├── all  
    │   │   ├── human_annot  
    │   │   ├── machine_annot  
    │   │   ├── skeleton.txt  
    │   │   └── subject.txt  
    │   └── images  
    │       ├── train  
    │       │   └── Capture0 ~ Capture26  
    │       └── val
```

(continues on next page)

(continued from previous page)

		-- Capture0
		— test
		-- Capture0 ~ Capture7

OVERVIEW

- Number of checkpoints: 336
- Number of configs: 356
- Number of papers: 83
 - ALGORITHM: 30
 - BACKBONE: 15
 - DATASET: 35
 - OTHERS: 3

For supported datasets, see [datasets overview](#).

17.1 Animal

- Number of checkpoints: 43
- Number of configs: 43
- Number of papers: 9
 - [ALGORITHM] Deep High-Resolution Representation Learning for Human Pose Estimation (Topdown Heatmap + Hrnet on Macaque , Topdown Heatmap + Hrnet on Horse10 , Topdown Heatmap + Hrnet on Atrw , Topdown Heatmap + Hrnet on Ap10k , Topdown Heatmap + Hrnet on Animalpose)
 - [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Zebra , Topdown Heatmap + Resnet on Macaque , Topdown Heatmap + Resnet on Locust , Topdown Heatmap + Resnet on Horse10 , Topdown Heatmap + Resnet on Fly , Topdown Heatmap + Resnet on Atrw , Topdown Heatmap + Resnet on Ap10k , Topdown Heatmap + Resnet on Animalpose)
 - [DATASET] Ap-10k: A Benchmark for Animal Pose Estimation in the Wild (Topdown Heatmap + Resnet on Ap10k , Topdown Heatmap + Hrnet on Ap10k)
 - [DATASET] Atrw: A Benchmark for Amur Tiger Re-Identification in the Wild (Topdown Heatmap + Resnet on Atrw , Topdown Heatmap + Hrnet on Atrw)
 - [DATASET] Cross-Domain Adaptation for Animal Pose Estimation (Topdown Heatmap + Resnet on Animalpose , Topdown Heatmap + Hrnet on Animalpose)
 - [DATASET] Deepposekit, a Software Toolkit for Fast and Robust Animal Pose Estimation Using Deep Learning (Topdown Heatmap + Resnet on Zebra , Topdown Heatmap + Resnet on Locust)
 - [DATASET] Fast Animal Pose Estimation Using Deep Neural Networks (Topdown Heatmap + Resnet on Fly)

- [DATASET] Macaquepose: A Novel ‘In the Wild’ macaque Monkey Pose Dataset for Markerless Motion Capture (Topdown Heatmap + Hrnet on Macaque , Topdown Heatmap + Resnet on Macaque)
- [DATASET] Pretraining Boosts Out-of-Domain Robustness for Pose Estimation (Topdown Heatmap + Hrnet on Horse10 , Topdown Heatmap + Resnet on Horse10)

17.2 Body(2D,Kpt,Sview,Img)

- Number of checkpoints: 185
- Number of configs: 193
- Number of papers: 45
 - [ALGORITHM] Associative Embedding: End-to-End Learning for Joint Detection and Grouping (Associative Embedding + Hrnet on MHP , Associative Embedding + Higherhrnet on Crowdpose , Associative Embedding + Hourglass + Ae on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Resnet on Coco , Associative Embedding + Higherhrnet on Coco , Associative Embedding + Hrnet on Coco , Associative Embedding + Mobilenetv2 on Coco , Associative Embedding + Higherhrnet + Udp on Coco , Associative Embedding + Higherhrnet on Aic , Associative Embedding + Hrnet on Aic)
 - [ALGORITHM] Bottom-Up Human Pose Estimation via Disentangled Keypoint Regression (Dekr + Hrnet on Crowdpose , Dekr + Hrnet on Coco)
 - [ALGORITHM] Contextual Instance Decoupling for Robust Multi-Person Pose Estimation (Cid + Hrnet on Coco)
 - [ALGORITHM] Convolutional Pose Machines (Topdown Heatmap + CPM on Mpii , Topdown Heatmap + CPM on JHMDB , Topdown Heatmap + CPM on Coco)
 - [ALGORITHM] Deep High-Resolution Representation Learning for Human Pose Estimation (Topdown Heatmap + Hrnet on Posetrack18 , Topdown Heatmap + Hrnet on Ochuman , Topdown Heatmap + Hrnet + Dark on Mpii , Topdown Heatmap + Hrnet on Mpii , Associative Embedding + Hrnet on MHP , Topdown Heatmap + Hrnet on H36m , Topdown Heatmap + Hrnet on Crowdpose , Dekr + Hrnet on Crowdpose , Topdown Heatmap + Hrnet + Dark on Coco , Topdown Heatmap + Hrnet + Udp on Coco , Topdown Heatmap + Hrnet + Augmentation on Coco , Topdown Heatmap + Hrnet on Coco , Topdown Heatmap + Hrnet + Fp16 on Coco , Dekr + Hrnet on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Hrnet on Coco , Topdown Heatmap + Hrnet on Aic , Associative Embedding + Hrnet on Aic)
 - [ALGORITHM] Deeppose: Human Pose Estimation via Deep Neural Networks (Deeppose + Resnet + Rle on Mpii , Deeppose + Resnet on Mpii , Deeppose + Resnet + Rle on Coco , Deeppose + Resnet on Coco)
 - [ALGORITHM] Distribution-Aware Coordinate Representation for Human Pose Estimation (Topdown Heatmap + Hrnet + Dark on Mpii , Topdown Heatmap + Hrnet + Dark on Coco , Topdown Heatmap + Resnet + Dark on Coco)
 - [ALGORITHM] Higherhrnet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation (Associative Embedding + Higherhrnet on Crowdpose , Associative Embedding + Higherhrnet on Coco , Associative Embedding + Higherhrnet + Udp on Coco , Associative Embedding + Higherhrnet on Aic)
 - [ALGORITHM] Hrformer: High-Resolution Vision Transformer for Dense Predict (Topdown Heatmap + Hrformer on Coco)
 - [ALGORITHM] Human Pose Regression With Residual Log-Likelihood Estimation (Deeppose + Resnet + Rle on Mpii , Deeppose + Resnet + Rle on Coco)

- [ALGORITHM] Improving Convolutional Networks With Self-Calibrated Convolutions (Topdown Heatmap + Snet on Mpii , Topdown Heatmap + Snet on Coco)
- [ALGORITHM] Learning Delicate Local Representations for Multi-Person Pose Estimation (Topdown Heatmap + RSN on Coco)
- [ALGORITHM] Lite-Hrnet: A Lightweight High-Resolution Network (Topdown Heatmap + Litehrnet on Mpii , Topdown Heatmap + Litehrnet on Coco)
- [ALGORITHM] Rethinking on Multi-Stage Networks for Human Pose Estimation (Topdown Heatmap + MSPN on Coco)
- [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Posetrack18 , Topdown Heatmap + Resnet on Ochuman , Topdown Heatmap + Resnet + Mpii on Mpii_trb , Topdown Heatmap + Resnet on Mpii , Topdown Heatmap + Resnet on MHP , Topdown Heatmap + Resnet on JHMDB , Topdown Heatmap + Resnet on Crowdpose , Topdown Heatmap + Resnet on Coco , Topdown Heatmap + Swin on Coco , Topdown Heatmap + Resnet + Dark on Coco , Topdown Heatmap + Resnet + Fp16 on Coco , Topdown Heatmap + Resnet on Aic)
- [ALGORITHM] Stacked Hourglass Networks for Human Pose Estimation (Topdown Heatmap + Hourglass on Mpii , Topdown Heatmap + Hourglass on Coco)
- [ALGORITHM] The Devil Is in the Details: Delving Into Unbiased Data Processing for Human Pose Estimation (Topdown Heatmap + Hrnet + Udp on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Higherhrnet + Udp on Coco)
- [ALGORITHM] Vipnas: Efficient Video Pose Estimation via Neural Architecture Search (Topdown Heatmap + Vipnas on Coco)
- [BACKBONE] Aggregated Residual Transformations for Deep Neural Networks (Topdown Heatmap + Resnext on Mpii , Topdown Heatmap + Resnext on Coco)
- [BACKBONE] Associative Embedding: End-to-End Learning for Joint Detection and Grouping (Associative Embedding + Hrnet on MHP , Associative Embedding + Higherhrnet on Crowdpose , Associative Embedding + Hourglass + Ae on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Resnet on Coco , Associative Embedding + Higherhrnet on Coco , Associative Embedding + Hrnet on Coco , Associative Embedding + Mobilenetv2 on Coco , Associative Embedding + Higherhrnet + Udp on Coco , Associative Embedding + Higherhrnet on Aic , Associative Embedding + Hrnet on Aic)
- [BACKBONE] Bag of Tricks for Image Classification With Convolutional Neural Networks (Topdown Heatmap + Resnetv1d on Mpii , Topdown Heatmap + Resnetv1d on Coco)
- [BACKBONE] Deep High-Resolution Representation Learning for Human Pose Estimation (Topdown Heatmap + Hrnet on Posetrack18 , Topdown Heatmap + Hrnet on Ochuman , Topdown Heatmap + Hrnet + Dark on Mpii , Topdown Heatmap + Hrnet on Mpii , Associative Embedding + Hrnet on MHP , Topdown Heatmap + Hrnet on H36m , Topdown Heatmap + Hrnet on Crowdpose , Dekr + Hrnet on Crowdpose , Topdown Heatmap + Hrnet + Dark on Coco , Topdown Heatmap + Hrnet + Udp on Coco , Topdown Heatmap + Hrnet + Augmentation on Coco , Topdown Heatmap + Hrnet on Coco , Topdown Heatmap + Hrnet + Fp16 on Coco , Dekr + Hrnet on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Hrnet on Coco , Topdown Heatmap + Hrnet on Aic , Associative Embedding + Hrnet on Aic)
- [BACKBONE] Deep Residual Learning for Image Recognition (Topdown Heatmap + Resnet on Posetrack18 , Topdown Heatmap + Resnet on Ochuman , Topdown Heatmap + Resnet + Mpii on Mpii_trb , Topdown Heatmap + Resnet on Mpii , Deeppose + Resnet + Rle on Mpii , Deeppose + Resnet on Mpii , Topdown Heatmap + Resnet on MHP , Topdown Heatmap + Resnet on JHMDB , Topdown Heatmap + Resnet on Crowdpose , Topdown Heatmap + Resnet on Coco , Topdown Heatmap + Resnet + Dark on Coco , Topdown Heatmap + Resnet + Fp16 on Coco , Deeppose + Resnet + Rle on Coco , Deeppose + Resnet on Coco , Associative Embedding + Resnet on Coco , Topdown Heatmap + Resnet on Aic)

- [BACKBONE] Imagenet Classification With Deep Convolutional Neural Networks (Topdown Heatmap + Alexnet on Coco)
- [BACKBONE] Mobilenetv2: Inverted Residuals and Linear Bottlenecks (Topdown Heatmap + Mobilenetv2 on Mpii , Topdown Heatmap + Mobilenetv2 on Coco , Associative Embedding + Mobilenetv2 on Coco)
- [BACKBONE] Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions (Topdown Heatmap + PVT on Coco)
- [BACKBONE] Resnest: Split-Attention Networks (Topdown Heatmap + Resnest on Coco)
- [BACKBONE] Shufflenet V2: Practical Guidelines for Efficient CNN Architecture Design (Topdown Heatmap + Shufflenetv2 on Mpii , Topdown Heatmap + Shufflenetv2 on Coco)
- [BACKBONE] Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices (Topdown Heatmap + Shufflenetv1 on Mpii , Topdown Heatmap + Shufflenetv1 on Coco)
- [BACKBONE] Squeeze-and-Excitation Networks (Topdown Heatmap + Seresnet on Mpii , Topdown Heatmap + Seresnet on Coco)
- [BACKBONE] Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows (Topdown Heatmap + Swin on Coco)
- [BACKBONE] Very Deep Convolutional Networks for Large-Scale Image Recognition (Topdown Heatmap + VGG on Coco)
- [DATASET] 2d Human Pose Estimation: New Benchmark and State of the Art Analysis (Topdown Heatmap + Hourglass on Mpii , Topdown Heatmap + Resnetv1d on Mpii , Topdown Heatmap + Seresnet on Mpii , Topdown Heatmap + Snet on Mpii , Topdown Heatmap + Hrnet + Dark on Mpii , Topdown Heatmap + Shufflenetv1 on Mpii , Topdown Heatmap + Resnext on Mpii , Topdown Heatmap + Litehrnet on Mpii , Topdown Heatmap + Shufflenetv2 on Mpii , Topdown Heatmap + CPM on Mpii , Topdown Heatmap + Resnet on Mpii , Topdown Heatmap + Mobilenetv2 on Mpii , Topdown Heatmap + Hrnet on Mpii , Deeppose + Resnet + Rle on Mpii , Deeppose + Resnet on Mpii)
- [DATASET] Ai Challenger: A Large-Scale Dataset for Going Deeper in Image Understanding (Topdown Heatmap + Resnet on Aic , Topdown Heatmap + Hrnet on Aic , Associative Embedding + Higherhrnet on Aic , Associative Embedding + Hrnet on Aic)
- [DATASET] Crowdpose: Efficient Crowded Scenes Pose Estimation and a New Benchmark (Topdown Heatmap + Resnet on Crowdpose , Topdown Heatmap + Hrnet on Crowdpose , Dekr + Hrnet on Crowdpose , Associative Embedding + Higherhrnet on Crowdpose)
- [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (Topdown Heatmap + Hrnet on H36m)
- [DATASET] Microsoft Coco: Common Objects in Context (Cid + Hrnet on Coco , Topdown Heatmap + CPM on Coco , Topdown Heatmap + Vipnas on Coco , Topdown Heatmap + Hrnet + Dark on Coco , Topdown Heatmap + Hrnet + Udp on Coco , Topdown Heatmap + Alexnet on Coco , Topdown Heatmap + Resnet on Coco , Topdown Heatmap + PVT on Coco , Topdown Heatmap + Resnest on Coco , Topdown Heatmap + Hrformer on Coco , Topdown Heatmap + Snet on Coco , Topdown Heatmap + Litehrnet on Coco , Topdown Heatmap + Swin on Coco , Topdown Heatmap + Hrnet + Augmentation on Coco , Topdown Heatmap + Shufflenetv2 on Coco , Topdown Heatmap + VGG on Coco , Topdown Heatmap + Resnet + Dark on Coco , Topdown Heatmap + Resnext on Coco , Topdown Heatmap + Hrnet on Coco , Topdown Heatmap + Hrnet + Fp16 on Coco , Topdown Heatmap + MSPN on Coco , Topdown Heatmap + Shufflenetv1 on Coco , Topdown Heatmap + Mobilenetv2 on Coco , Topdown Heatmap + Resnet + Fp16 on Coco , Topdown Heatmap + RSN on Coco , Topdown Heatmap + Hourglass on Coco , Topdown Heatmap + Resnetv1d on Coco , Topdown Heatmap + Seresnet on Coco , Dekr + Hrnet on Coco , Deeppose + Resnet + Rle on Coco , Deeppose + Resnet on Coco , Associative Embedding + Hourglass + Ae on Coco , Associative Embedding + Hrnet + Udp on Coco , Associative Embedding + Resnet on Coco , Associative

- Embedding + Higherhrnet on Coco , Associative Embedding + Hrnet on Coco , Associative Embedding + Mobilenetv2 on Coco , Associative Embedding + Higherhrnet + Udp on Coco)
- [DATASET] Pose2seg: Detection Free Human Instance Segmentation (Topdown Heatmap + Hrnet on Ochuman , Topdown Heatmap + Resnet on Ochuman)
 - [DATASET] Posetrack: A Benchmark for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Posetrack18 , Topdown Heatmap + Hrnet on Posetrack18)
 - [DATASET] Towards Understanding Action Recognition (Topdown Heatmap + Resnet on JHMDB , Topdown Heatmap + CPM on JHMDB)
 - [DATASET] Trb: A Novel Triplet Representation for Understanding 2d Human Body (Topdown Heatmap + Resnet + Mpii on Mpii_trb)
 - [DATASET] Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and a New Benchmark for Multi-Human Parsing (Topdown Heatmap + Resnet on MHP , Associative Embedding + Hrnet on MHP)
 - [OTHERS] Albumentations: Fast and Flexible Image Augmentations (Topdown Heatmap + Hrnet + Augmentation on Coco)
 - [OTHERS] Feature Pyramid Networks for Object Detection (Topdown Heatmap + Swin on Coco)
 - [OTHERS] Mixed Precision Training (Topdown Heatmap + Hrnet + Fp16 on Coco , Topdown Heatmap + Resnet + Fp16 on Coco)

17.3 Body(2D,Kpt,Sview,Vid)

- Number of checkpoints: 3
- Number of configs: 2
- Number of papers: 4
 - [ALGORITHM] Deep High-Resolution Representation Learning for Human Pose Estimation (Posewarper + Hrnet + Posetrack18 on Posetrack18)
 - [ALGORITHM] Learning Temporal Pose Estimation From Sparsely Labeled Videos (Posewarper + Hrnet + Posetrack18 on Posetrack18)
 - [DATASET] Microsoft Coco: Common Objects in Context (Posewarper + Hrnet + Posetrack18 on Posetrack18)
 - [DATASET] Posetrack: A Benchmark for Human Pose Estimation and Tracking (Posewarper + Hrnet + Posetrack18 on Posetrack18)

17.4 Body(3D,Kpt,Mview,Img)

- Number of checkpoints: 5
- Number of configs: 5
- Number of papers: 3
 - [ALGORITHM] Voxelpose: Towards Multi-Camera 3d Human Pose Estimation in Wild Environment (Voxelpose + Voxelpose on Shelf , Voxelpose + Voxelpose + Prn64x64x64 + Cpn80x80x20 + Panoptic on Panoptic , Voxelpose + Voxelpose on Campus)

- [DATASET] Panoptic Studio: A Massively Multiview System for Social Motion Capture (Voxelpose + Voxelpose + Prn64x64x64 + Cpn80x80x20 + Panoptic on Panoptic)
- [DATASET] {3d (Voxelpose + Voxelpose on Shelf , Voxelpose + Voxelpose on Campus)

17.5 Body(3D,Kpt,Sview,Img)

- Number of checkpoints: 2
- Number of configs: 2
- Number of papers: 3
 - [ALGORITHM] A Simple Yet Effective Baseline for 3d Human Pose Estimation (Pose Lift + Simplebaseline3d on Mpi_inf_3dhp , Pose Lift + Simplebaseline3d on H36m)
 - [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (Pose Lift + Simplebaseline3d on H36m)
 - [DATASET] Monocular 3d Human Pose Estimation in the Wild Using Improved CNN Supervision (Pose Lift + Simplebaseline3d on Mpi_inf_3dhp)

17.6 Body(3D,Kpt,Sview,Vid)

- Number of checkpoints: 8
- Number of configs: 8
- Number of papers: 3
 - [ALGORITHM] 3d Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training (Video Pose Lift + Videopose3d on Mpi_inf_3dhp , Video Pose Lift + Videopose3d on H36m)
 - [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (Video Pose Lift + Videopose3d on H36m)
 - [DATASET] Monocular 3d Human Pose Estimation in the Wild Using Improved CNN Supervision (Video Pose Lift + Videopose3d on Mpi_inf_3dhp)

17.7 Body(3D,Mesh,Sview,Img)

- Number of checkpoints: 1
- Number of configs: 1
- Number of papers: 3
 - [ALGORITHM] End-to-End Recovery of Human Shape and Pose (HMR + Resnet on Mixed)
 - [BACKBONE] Deep Residual Learning for Image Recognition (HMR + Resnet on Mixed)
 - [DATASET] Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments (HMR + Resnet on Mixed)

17.8 Face

- Number of checkpoints: 16
- Number of configs: 16
- Number of papers: 16
 - [ALGORITHM] Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression (Topdown Heatmap + Hrnetv2 + Awing on WFLW)
 - [ALGORITHM] Deep High-Resolution Representation Learning for Visual Recognition (Topdown Heatmap + Hrnetv2 on WFLW , Topdown Heatmap + Hrnetv2 + Dark on WFLW , Topdown Heatmap + Hrnetv2 + Awing on WFLW , Topdown Heatmap + Hrnetv2 on Cofw , Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Hrnetv2 on Aflw , Topdown Heatmap + Hrnetv2 + Dark on Aflw , Topdown Heatmap + Hrnetv2 on 300w)
 - [ALGORITHM] Deeppose: Human Pose Estimation via Deep Neural Networks (Deeppose + Resnet + Wingloss on WFLW , Deeppose + Resnet + Softwingloss on WFLW , Deeppose + Resnet on WFLW)
 - [ALGORITHM] Distribution-Aware Coordinate Representation for Human Pose Estimation (Topdown Heatmap + Hrnetv2 + Dark on WFLW , Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Hrnetv2 + Dark on Aflw)
 - [ALGORITHM] Improving Convolutional Networks With Self-Calibrated Convolutions (Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face)
 - [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face)
 - [ALGORITHM] Stacked Hourglass Networks for Human Pose Estimation (Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face)
 - [ALGORITHM] Structure-Coherent Deep Feature Learning for Robust Face Alignment (Deeppose + Resnet + Softwingloss on WFLW)
 - [ALGORITHM] Wing Loss for Robust Facial Landmark Localisation With Convolutional Neural Networks (Deeppose + Resnet + Wingloss on WFLW)
 - [BACKBONE] Deep Residual Learning for Image Recognition (Deeppose + Resnet + Wingloss on WFLW , Deeppose + Resnet + Softwingloss on WFLW , Deeppose + Resnet on WFLW , Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face)
 - [BACKBONE] Mobilenetv2: Inverted Residuals and Linear Bottlenecks (Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_face)
 - [DATASET] 300 Faces in-the-Wild Challenge: Database and Results (Topdown Heatmap + Hrnetv2 on 300w)
 - [DATASET] Annotated Facial Landmarks in the Wild: A Large-Scale, Real-World Database for Facial Landmark Localization (Topdown Heatmap + Hrnetv2 on Aflw , Topdown Heatmap + Hrnetv2 + Dark on Aflw)
 - [DATASET] Look at Boundary: A Boundary-Aware Face Alignment Algorithm (Topdown Heatmap + Hrnetv2 on WFLW , Topdown Heatmap + Hrnetv2 + Dark on WFLW , Topdown Heatmap + Hrnetv2 + Awing on WFLW , Deeppose + Resnet + Wingloss on WFLW , Deeppose + Resnet + Softwingloss on WFLW , Deeppose + Resnet on WFLW)
 - [DATASET] Robust Face Landmark Estimation Under Occlusion (Topdown Heatmap + Hrnetv2 on Cofw)

- [DATASET] Whole-Body Human Pose Estimation in the Wild (Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face , Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_face)

17.9 Fashion

- Number of checkpoints: 19
- Number of configs: 19
- Number of papers: 6
 - [ALGORITHM] Deeppose: Human Pose Estimation via Deep Neural Networks (Deeppose + Resnet on Deepfashion)
 - [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Deepfashion2 , Topdown Heatmap + Resnet on Deepfashion)
 - [BACKBONE] Deep Residual Learning for Image Recognition (Topdown Heatmap + Resnet on Deepfashion2 , Topdown Heatmap + Resnet on Deepfashion , Deeppose + Resnet on Deepfashion)
 - [DATASET] A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images (Topdown Heatmap + Resnet on Deepfashion2)
 - [DATASET] Deepfashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations (Topdown Heatmap + Resnet on Deepfashion , Deeppose + Resnet on Deepfashion)
 - [DATASET] Fashion Landmark Detection in the Wild (Topdown Heatmap + Resnet on Deepfashion , Deeppose + Resnet on Deepfashion)

17.10 Hand(2D,Kpt,Rgb,Img)

- Number of checkpoints: 29
- Number of configs: 39
- Number of papers: 16
 - [ALGORITHM] Deep High-Resolution Representation Learning for Visual Recognition (Topdown Heatmap + Hrnetv2 on Rhd2d , Topdown Heatmap + Hrnetv2 + Dark on Rhd2d , Topdown Heatmap + Hrnetv2 + Udp on Rhd2d , Topdown Heatmap + Hrnetv2 on Panoptic2d , Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d , Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d , Topdown Heatmap + Hrnetv2 + Udp on Onehand10k , Topdown Heatmap + Hrnetv2 on Onehand10k , Topdown Heatmap + Hrnetv2 + Dark on Onehand10k , Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand)
 - [ALGORITHM] Deeppose: Human Pose Estimation via Deep Neural Networks (Deeppose + Resnet on Rhd2d , Deeppose + Resnet on Panoptic2d , Deeppose + Resnet on Onehand10k)
 - [ALGORITHM] Distribution-Aware Coordinate Representation for Human Pose Estimation (Topdown Heatmap + Hrnetv2 + Dark on Rhd2d , Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d , Topdown Heatmap + Hrnetv2 + Dark on Onehand10k , Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand)

- [ALGORITHM] Improving Convolutional Networks With Self-Calibrated Convolutions (Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_hand)
- [ALGORITHM] Lite-Hrnet: A Lightweight High-Resolution Network (Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand)
- [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Rhd2d , Topdown Heatmap + Resnet on Panoptic2d , Topdown Heatmap + Resnet on Onehand10k , Topdown Heatmap + Resnet on Interhand2d , Topdown Heatmap + Resnet on Freihand2d , Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand)
- [ALGORITHM] Stacked Hourglass Networks for Human Pose Estimation (Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand)
- [ALGORITHM] The Devil Is in the Details: Delving Into Unbiased Data Processing for Human Pose Estimation (Topdown Heatmap + Hrnetv2 + Udp on Rhd2d , Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d , Topdown Heatmap + Hrnetv2 + Udp on Onehand10k)
- [BACKBONE] Deep Residual Learning for Image Recognition (Topdown Heatmap + Resnet on Rhd2d , Deeppose + Resnet on Rhd2d , Topdown Heatmap + Resnet on Panoptic2d , Deeppose + Resnet on Panoptic2d , Topdown Heatmap + Resnet on Onehand10k , Deeppose + Resnet on Onehand10k , Topdown Heatmap + Resnet on Interhand2d , Topdown Heatmap + Resnet on Freihand2d , Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand)
- [BACKBONE] Mobilenetv2: Inverted Residuals and Linear Bottlenecks (Topdown Heatmap + Mobilenetv2 on Rhd2d , Topdown Heatmap + Mobilenetv2 on Panoptic2d , Topdown Heatmap + Mobilenetv2 on Onehand10k , Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_hand)
- [DATASET] Freihand: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images (Topdown Heatmap + Resnet on Freihand2d)
- [DATASET] Hand Keypoint Detection in Single Images Using Multiview Bootstrapping (Topdown Heatmap + Hrnetv2 on Panoptic2d , Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d , Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d , Topdown Heatmap + Resnet on Panoptic2d , Topdown Heatmap + Mobilenetv2 on Panoptic2d , Deeppose + Resnet on Panoptic2d)
- [DATASET] Interhand2.6m: A Dataset and Baseline for 3d Interacting Hand Pose Estimation From a Single RGB Image (Topdown Heatmap + Resnet on Interhand2d)
- [DATASET] Learning to Estimate 3d Hand Pose From Single RGB Images (Topdown Heatmap + Hrnetv2 on Rhd2d , Topdown Heatmap + Hrnetv2 + Dark on Rhd2d , Topdown Heatmap + Hrnetv2 + Udp on Rhd2d , Topdown Heatmap + Resnet on Rhd2d , Topdown Heatmap + Mobilenetv2 on Rhd2d , Deeppose + Resnet on Rhd2d)
- [DATASET] Mask-Pose Cascaded CNN for 2d Hand Pose Estimation From Single Color Image (Topdown Heatmap + Hrnetv2 + Udp on Onehand10k , Topdown Heatmap + Mobilenetv2 on Onehand10k , Topdown Heatmap + Hrnetv2 on Onehand10k , Topdown Heatmap + Resnet on Onehand10k , Topdown Heatmap + Hrnetv2 + Dark on Onehand10k , Deeppose + Resnet on Onehand10k)
- [DATASET] Whole-Body Human Pose Estimation in the Wild (Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_hand , Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand)

17.11 Hand(3D,Kpt,Rgb,Img)

- Number of checkpoints: 1
- Number of configs: 2
- Number of papers: 3
 - [ALGORITHM] Interhand2.6m: A Dataset and Baseline for 3d Interacting Hand Pose Estimation From a Single RGB Image (Internet + Internet on Interhand3d)
 - [BACKBONE] Deep Residual Learning for Image Recognition (Internet + Internet on Interhand3d)
 - [DATASET] Interhand2.6m: A Dataset and Baseline for 3d Interacting Hand Pose Estimation From a Single RGB Image (Internet + Internet on Interhand3d)

17.12 Hand(Gesture,Rgbd,Vid)

- Number of checkpoints: 3
- Number of configs: 4
- Number of papers: 3
 - [ALGORITHM] Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With Multimodal Training (Mtut + I3d on Nvgesture)
 - [BACKBONE] Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset ()
 - [DATASET] Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3d Convolutional Neural Network (Mtut + I3d on Nvgesture)

17.13 Wholebody

- Number of checkpoints: 22
- Number of configs: 22
- Number of papers: 9
 - [ALGORITHM] Associative Embedding: End-to-End Learning for Joint Detection and Grouping (Associative Embedding + Higherhrnet on Coco-Wholebody , Associative Embedding + Hrnet on Coco-Wholebody)
 - [ALGORITHM] Deep High-Resolution Representation Learning for Human Pose Estimation (Topdown Heatmap + Hrnet + Dark on Halpe , Topdown Heatmap + Hrnet + Dark on Coco-Wholebody , Topdown Heatmap + Hrnet on Coco-Wholebody , Associative Embedding + Hrnet on Coco-Wholebody)
 - [ALGORITHM] Distribution-Aware Coordinate Representation for Human Pose Estimation (Topdown Heatmap + Hrnet + Dark on Halpe , Topdown Heatmap + Hrnet + Dark on Coco-Wholebody , Topdown Heatmap + Vipnas + Dark on Coco-Wholebody)
 - [ALGORITHM] Higherhrnet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation (Associative Embedding + Higherhrnet on Coco-Wholebody)
 - [ALGORITHM] Not All Tokens Are Equal: Human-Centric Visual Analysis via Token Clustering Transformer (Topdown Heatmap + Tcformer on Coco-Wholebody)

- [ALGORITHM] Simple Baselines for Human Pose Estimation and Tracking (Topdown Heatmap + Resnet on Coco-Wholebody)
- [ALGORITHM] Vipnas: Efficient Video Pose Estimation via Neural Architecture Search (Topdown Heatmap + Vipnas + Dark on Coco-Wholebody , Topdown Heatmap + Vipnas on Coco-Wholebody)
- [DATASET] Pastanet: Toward Human Activity Knowledge Engine (Topdown Heatmap + Hrnet + Dark on Halpe)
- [DATASET] Whole-Body Human Pose Estimation in the Wild (Topdown Heatmap + Hrnet + Dark on Coco-Wholebody , Topdown Heatmap + Vipnas + Dark on Coco-Wholebody , Topdown Heatmap + Vipnas on Coco-Wholebody , Topdown Heatmap + Tformer on Coco-Wholebody , Topdown Heatmap + Hrnet on Coco-Wholebody , Topdown Heatmap + Resnet on Coco-Wholebody , Associative Embedding + Higherhrnet on Coco-Wholebody , Associative Embedding + Hrnet on Coco-Wholebody)

18.1 Animalpose Dataset

18.1.1 Topdown Heatmap + Hrnet on Animalpose

```
@inproceedings{sun2019deep,  
  title={Deep high-resolution representation learning for human pose estimation},  
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_  
↪ recognition},  
  pages={5693--5703},  
  year={2019}  
}
```

```
@InProceedings{Cao_2019_ICCV,  
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,  
↪ Cewu and Tai, Yu-Wing},  
  title = {Cross-Domain Adaptation for Animal Pose Estimation},  
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},  
  month = {October},  
  year = {2019}  
}
```

Results on AnimalPose validation set (1117 instances)

18.1.2 Topdown Heatmap + Resnet on Animalpose

```
@inproceedings{xiao2018simple,  
  title={Simple baselines for human pose estimation and tracking},  
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},  
  booktitle={Proceedings of the European conference on computer vision (ECCV)},  
  pages={466--481},  
  year={2018}  
}
```

```
@InProceedings{Cao_2019_ICCV,  
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,  
↪ Cewu and Tai, Yu-Wing},
```

(continues on next page)

(continued from previous page)

```

    title = {Cross-Domain Adaptation for Animal Pose Estimation},
    booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
    month = {October},
    year = {2019}
}

```

Results on AnimalPose validation set (1117 instances)

18.2 Ap10k Dataset

18.2.1 Topdown Heatmap + Hrnet on Ap10k

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng
↪Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```

Results on AP-10K validation set

18.2.2 Topdown Heatmap + Resnet on Ap10k

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng
↪Tao},
  year={2021},

```

(continues on next page)

(continued from previous page)

```

eprint={2108.12617},
archivePrefix={arXiv},
primaryClass={cs.CV}
}

```

Results on AP-10K validation set

18.3 Atrw Dataset

18.3.1 Topdown Heatmap + Hrnet on Atrw

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}

```

Results on ATRW validation set

18.3.2 Topdown Heatmap + Resnet on Atrw

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}

```

Results on ATRW validation set

18.4 Fly Dataset

18.4.1 Topdown Heatmap + Resnet on Fly

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{pereira2019fast,
  title={Fast animal pose estimation using deep neural networks},
  author={Pereira, Talmo D and Aldarondo, Diego E and Willmore, Lindsay and Kislin,
  ↪Mikhail and Wang, Samuel S-H and Murthy, Mala and Shaevitz, Joshua W},
  journal={Nature methods},
  volume={16},
  number={1},
  pages={117--125},
  year={2019},
  publisher={Nature Publishing Group}
}
```

Results on Vinegar Fly test set

18.5 Horse10 Dataset

18.5.1 Topdown Heatmap + Resnet on Horse10

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekgonul,
  ↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
  ↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

18.5.2 Topdown Heatmap + Hrnet on Horse10

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekogonul,
↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

18.6 Locust Dataset

18.6.1 Topdown Heatmap + Resnet on Locust

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation
↪using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,
↪Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

Results on Desert Locust test set

18.7 Macaque Dataset

18.7.1 Topdown Heatmap + Resnet on Macaque

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel 'in the wild'macaque monkey pose dataset for markerless_
↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,_
↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi_
↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

18.7.2 Topdown Heatmap + Hrnet on Macaque

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel 'in the wild'macaque monkey pose dataset for markerless_
↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,_
↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi_
↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

18.8 Zebra Dataset

18.8.1 Topdown Heatmap + Resnet on Zebra

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation,
  ↪ using deep learning},
  ↪ author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,
  ↪ Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

Results on Grévy's Zebra test set

BODY(2D,KPT,SVIEW,IMG)

19.1 Aic Dataset

19.1.1 Associative Embedding + Hrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.1.2 Associative Embedding + Higherhrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and ↵
↵ Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↵,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.1.3 Topdown Heatmap + Hrnet on Aic

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and ↵
↵ Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↵,
  journal={arXiv preprint arXiv:1711.06475},
```

(continues on next page)

(continued from previous page)

```

    year={2017}
}

```

Results on AIC val set with ground-truth bounding boxes

19.1.4 Topdown Heatmap + Resnet on Aic

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}

```

Results on AIC val set with ground-truth bounding boxes

19.2 Coco Dataset

19.2.1 Associative Embedding + Higherhrnet + Udp on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

19.2.2 Associative Embedding + Mobilenetv2 on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and_
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={4510--4520},
}
```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.2.3 Associative Embedding + Hrnet on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.2.4 Associative Embedding + Higherhrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.2.5 Associative Embedding + Resnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵ recognition},
  pages={770--778},
}
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.2.6 Associative Embedding + Hrnet + Udp on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},

```

(continues on next page)

(continued from previous page)

```

booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

19.2.7 Associative Embedding + Hourglass + Ae on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.2.8 Deeppose + Resnet on Coco

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.9 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.10 Dekr + Hrnet on Coco

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

The results of models provided by the authors on COCO val2017 using the same evaluation protocol

The discrepancy between these results and that shown in paper is attributed to the differences in implementation details in evaluation process.

19.2.11 Topdown Heatmap + Seresnet on Coco

```
@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={7132--7141},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that * means without imagenet pre-training.

19.2.12 Topdown Heatmap + Resnetv1d on Coco

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.13 Topdown Heatmap + Hourglass on Coco

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↳ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.14 Topdown Heatmap + RSN on Coco

```
@misc{cai2020learning,
  title={Learning Delicate Local Representations for Multi-Person Pose Estimation},
  author={Yuanhao Cai and Zhicheng Wang and Zhengxiong Luo and Binyi Yin and Angang Du
↳ and Haoqian Wang and Xinyu Zhou and Erjin Zhou and Xiangyu Zhang and Jian Sun},
  year={2020},
  eprint={2003.04030},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↳ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.15 Topdown Heatmap + Resnet + Fp16 on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.16 Topdown Heatmap + Mobilenetv2 on Coco

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.17 Topdown Heatmap + Shufflenetv1 on Coco

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile
  ↪devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.18 Topdown Heatmap + MSPN on Coco

```
@article{li2019rethinking,
  title={Rethinking on Multi-Stage Networks for Human Pose Estimation},
  author={Li, Wenbo and Wang, Zhicheng and Yin, Binyi and Peng, Qixiang and Du, Yuming
  ↪and Xiao, Tianzi and Yu, Gang and Lu, Hongtao and Wei, Yichen and Sun, Jian},
  journal={arXiv preprint arXiv:1901.00148},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

(continues on next page)

(continued from previous page)

```

booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.19 Topdown Heatmap + Hrnet + Fp16 on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory_
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and_
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.20 Topdown Heatmap + Hrnet on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},

```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.21 Topdown Heatmap + Resnext on Coco

```

@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{'a}r, Piotr and Tu, Zhuowen and He,
↪Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1492--1500},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.22 Topdown Heatmap + Resnet + Dark on Coco

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.23 Topdown Heatmap + VGG on Coco

```
@article{simonyan2014very,
  title={Very deep convolutional networks for large-scale image recognition},
  author={Simonyan, Karen and Zisserman, Andrew},
  journal={arXiv preprint arXiv:1409.1556},
  year={2014}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.24 Topdown Heatmap + Shufflenetv2 on Coco

```
@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.25 Topdown Heatmap + Hrnet + Augmentation on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{buslaev2020albumentations,
  title={Albumentations: fast and flexible image augmentations},
  author={Buslaev, Alexander and Iglovikov, Vladimir I and Khvedchenya, Eugene and
  Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A},
  journal={Information},
  volume={11},
  number={2},
  pages={125},
  year={2020},
  publisher={Multidisciplinary Digital Publishing Institute}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

(continues on next page)

(continued from previous page)

```

    year={2014},
    organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.26 Topdown Heatmap + Swin on Coco

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{liu2021swin,
  title={Swin transformer: Hierarchical vision transformer using shifted windows},
  author={Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang,
↪Zheng and Lin, Stephen and Guo, Baining},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={10012--10022},
  year={2021}
}

```

```

@inproceedings{lin2017feature,
  title={Feature pyramid networks for object detection},
  author={Lin, Tsung-Yi and Doll{\`a}r, Piotr and Girshick, Ross and He, Kaiming and
↪Hariharan, Bharath and Belongie, Serge},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2117--2125},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.27 Topdown Heatmap + Litehrnet on Coco

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.28 Topdown Heatmap + Scnet on Coco

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.29 Topdown Heatmap + Hrformer on Coco

```
@article{yuan2021hrformer,
  title={HRFormer: High-Resolution Vision Transformer for Dense Predict},
  author={Yuan, Yuhui and Fu, Rao and Huang, Lang and Lin, Weihong and Zhang, Chao and
↪Chen, Xilin and Wang, Jingdong},
  journal={Advances in Neural Information Processing Systems},
  volume={34},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.30 Topdown Heatmap + Resnest on Coco

```
@article{zhang2020resnest,
  title={ResNeSt: Split-Attention Networks},
  author={Zhang, Hang and Wu, Chongruo and Zhang, Zhongyue and Zhu, Yi and Zhang, Zhi
↪and Lin, Haibin and Sun, Yue and He, Tong and Muller, Jonas and Manmatha, R. and Li,
↪Mu and Smola, Alexander},
  journal={arXiv preprint arXiv:2004.08955},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.31 Topdown Heatmap + PVT on Coco

```
@inproceedings{wang2021pyramid,
  title={Pyramid vision transformer: A versatile backbone for dense prediction without
↪ convolutions},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao
↪ and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={568--578},
  year={2021}
}
```

```
@article{wang2022pvt,
  title={PVT v2: Improved baselines with Pyramid Vision Transformer},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao
↪ and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  journal={Computational Visual Media},
  pages={1--10},
  year={2022},
  publisher={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.32 Topdown Heatmap + Resnet on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={770--778},
}
```

(continues on next page)

(continued from previous page)

```

year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.33 Topdown Heatmap + Alexnet on Coco

```

@inproceedings{krizhevsky2012imagenet,
  title={Imagenet classification with deep convolutional neural networks},
  author={Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E},
  booktitle={Advances in neural information processing systems},
  pages={1097--1105},
  year={2012}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.34 Topdown Heatmap + Hrnet + Udp on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of [DARK](#).

19.2.35 Topdown Heatmap + Hrnet + Dark on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
}
```

(continues on next page)

(continued from previous page)

```
organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.36 Topdown Heatmap + Vipnas on Coco

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.37 Topdown Heatmap + CPM on Coco

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪ Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

19.2.38 Cid + Hrnet on Coco

```
@InProceedings{Wang_2022_CVPR,
  author    = {Wang, Dongkai and Zhang, Shiliang},
  title     = {Contextual Instance Decoupling for Robust Multi-Person Pose Estimation},
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition (CVPR)},
  month     = {June},
  year      = {2022},
  pages     = {11060-11068}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

19.3 Crowdpose Dataset

19.3.1 Associative Embedding + Higherhrnet on Crowdpose

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪ Lu, Cewu},
  (continues on next page)
```

(continued from previous page)

```

journal={arXiv preprint arXiv:1812.00324},
year={2018}
}

```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 2 scales ([2, 1]) are used

19.3.2 Dekr + Hrnet on Crowdpose

```

@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
  ↪Recognition},
  pages={14676--14686},
  year={2021}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
  ↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and_
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}

```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

19.3.3 Topdown Heatmap + Hrnet on Crowdpose

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test with YOLOv3 human detector

19.3.4 Topdown Heatmap + Resnet on Crowdpose

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test with YOLOv3 human detector

19.4 H36m Dataset

19.4.1 Topdown Heatmap + Hrnet on H36m

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M test set with ground truth 2D detections

19.5 JHMDB Dataset

19.5.1 Topdown Heatmap + CPM on JHMDB

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. NO test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

19.5.2 Topdown Heatmap + Resnet on JHMDB

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. NO test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

19.6 MHP Dataset

19.6.1 Associative Embedding + Hrnet on MHP

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
```

(continues on next page)

(continued from previous page)

```

    pages={2277--2287},
    year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}

```

Results on MHP v2.0 validation set without multi-scale test

Results on MHP v2.0 validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

19.6.2 Topdown Heatmap + Resnet on MHP

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},

```

(continues on next page)

(continued from previous page)

```
booktitle={Proceedings of the 26th ACM international conference on Multimedia},
pages={792--800},
year={2018}
}
```

Results on MHP v2.0 val set

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. Please be cautious if you use the results in papers.

19.7 Mpii Dataset

19.7.1 Deeppose + Resnet on Mpii

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.2 Deeppose + Resnet + Rle on Mpii

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.3 Topdown Heatmap + Hrnet on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.4 Topdown Heatmap + Mobilenetv2 on Mpii

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and ↪
  ↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.5 Topdown Heatmap + Resnet on Mpii

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={770--778},
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

19.7.6 Topdown Heatmap + CPM on Mpii

```

@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

19.7.7 Topdown Heatmap + Shufflenetv2 on Mpii

```

@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,

```

(continues on next page)

(continued from previous page)

```

title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
year = {2014},
month = {June}
}

```

Results on MPII val set

19.7.8 Topdown Heatmap + Litehrnet on Mpii

```

@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}

```

```

@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

19.7.9 Topdown Heatmap + Resnext on Mpii

```

@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{\`a}r, Piotr and Tu, Zhuowen and He,
↪Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1492--1500},
  year={2017}
}

```

```

@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

19.7.10 Topdown Heatmap + Shufflenetv1 on Mpii

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile
↪ devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.11 Topdown Heatmap + Hrnet + Dark on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
```

(continues on next page)

(continued from previous page)

```

year = {2014},
month = {June}
}

```

Results on MPII val set

19.7.12 Topdown Heatmap + Scnet on Mpii

```

@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

19.7.13 Topdown Heatmap + Seresnet on Mpii

```

@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={7132--7141},
  year={2018}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

Note that * means without imagenet pre-training.

19.7.14 Topdown Heatmap + Resnetv1d on Mpii

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.7.15 Topdown Heatmap + Hourglass on Mpii

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

19.8 Mpii_trb Dataset

19.8.1 Topdown Heatmap + Resnet + Mpii on Mpii_trb

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{duan2019trb,
  title={TRB: A Novel Triplet Representation for Understanding 2D Human Body},
  author={Duan, Haodong and Lin, Kwan-Yee and Jin, Sheng and Liu, Wentao and Qian, Chen
↪and Ouyang, Wanli},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={9479--9488},
  year={2019}
}
```

Results on MPII-TRB val set

19.9 Ochuman Dataset

19.9.1 Topdown Heatmap + Resnet on Ochuman

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
```

(continues on next page)

(continued from previous page)

```

    pages={770--778},
    year={2016}
}

```

```

@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi,
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern.
↪recognition},
  pages={889--898},
  year={2019}
}

```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

19.9.2 Topdown Heatmap + Hrnet on Ochuman

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern.
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi,
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern.
↪recognition},
  pages={889--898},
  year={2019}
}

```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

19.10 Posetrack18 Dataset

19.10.1 Topdown Heatmap + Hrnet on Posetrack18

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

19.10.2 Topdown Heatmap + Resnet on Posetrack18

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

BODY(2D,KPT,SVIEW,VID)

20.1 Posetrack18 Dataset

20.1.1 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,  
  title={Learning Temporal Pose Estimation from Sparsely Labeled Videos},  
  author={Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo, and  
    ↪Torresani, Lorenzo},  
  booktitle={Advances in Neural Information Processing Systems 33},  
  year={2019},  
}
```

```
@inproceedings{sun2019deep,  
  title={Deep high-resolution representation learning for human pose estimation},  
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
    ↪recognition},  
  pages={5693--5703},  
  year={2019}  
}
```

```
@inproceedings{andrilluka2018posetrack,  
  title={Posetrack: A benchmark for human pose estimation and tracking},  
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin, and  
    ↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},  
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern  
    ↪Recognition},  
  pages={5167--5176},  
  year={2018}  
}
```

```
@inproceedings{lin2014microsoft,  
  title={Microsoft coco: Common objects in context},  
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and  
    ↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},  
  booktitle={European conference on computer vision},  
  pages={740--755},  
  year={2014},  
}
```

(continues on next page)

(continued from previous page)

```
organization={Springer}
}
```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)¹.

¹ Please download the precomputed human bounding boxes on PoseTrack2018 val from `$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json` and place it here: `$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json` to be consistent with the [config](#). Please refer to DATA Preparation for more detail about data preparation.

BODY(3D,KPT,SVIEW,IMG)

21.1 H36m Dataset

21.1.1 Pose Lift + Simplebaseline3d on H36m

```
@inproceedings{martinez_2017_3dbaseline,  
  title={A simple yet effective baseline for 3d human pose estimation},  
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},  
  booktitle={ICCV},  
  year={2017}  
}
```

```
@article{h36m_pami,  
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↵  
↵Cristian},  
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↵  
↵in Natural Environments},  
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},  
  publisher = {IEEE Computer Society},  
  volume = {36},  
  number = {7},  
  pages = {1325-1339},  
  month = {jul},  
  year = {2014}  
}
```

Results on Human3.6M dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

21.2 Mpi_inf_3dhp Dataset

21.2.1 Pose Lift + Simplebaseline3d on Mpi_inf_3dhp

```
@inproceedings{martinez_2017_3dbaseline,
  title={A simple yet effective baseline for 3d human pose estimation},
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},
  booktitle={ICCV},
  year={2017}
}
```

```
@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and ↵
↵ Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
↵,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
  year = {2017},
  organization={IEEE},
  doi={10.1109/3dv.2017.00064},
}
```

Results on MPI-INF-3DHP dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

BODY(3D,KPT,SVIEW,VID)

22.1 H36m Dataset

22.1.1 Video Pose Lift + Videopose3d on H36m

```
@inproceedings{pavullo20193d,  
  title={3d human pose estimation in video with temporal convolutions and semi-  
↪supervised training},  
  author={Pavullo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli, ↪  
↪Michael},  
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↪  
↪Recognition},  
  pages={7753--7762},  
  year={2019}  
}
```

```
@article{h36m_pami,  
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪  
↪Cristian},  
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing ↪  
↪in Natural Environments},  
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},  
  publisher = {IEEE Computer Society},  
  volume = {36},  
  number = {7},  
  pages = {1325-1339},  
  month = {jul},  
  year = {2014}  
}
```

Results on Human3.6M dataset with ground truth 2D detections, supervised training

Results on Human3.6M dataset with CPN 2D detections¹, supervised training

Results on Human3.6M dataset with ground truth 2D detections, semi-supervised training

Results on Human3.6M dataset with CPN 2D detections¹, semi-supervised training

¹ CPN 2D detections are provided by [official repo](#). The reformatted version used in this repository can be downloaded from [train_detection](#) and [test_detection](#).

22.2 Mpi_inf_3dhp Dataset

22.2.1 Video Pose Lift + Videopose3d on Mpi_inf_3dhp

```
@inproceedings{pavullo20193d,  
  title={3d human pose estimation in video with temporal convolutions and semi-  
↪supervised training},  
  author={Pavullo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli, ↪  
↪Michael},  
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↪  
↪Recognition},  
  pages={7753--7762},  
  year={2019}  
}
```

```
@inproceedings{mono-3dhp2017,  
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and ↪  
↪Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},  
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision} ↪  
↪,  
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},  
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},  
  year = {2017},  
  organization={IEEE},  
  doi={10.1109/3dv.2017.00064},  
}
```

Results on MPI-INF-3DHP dataset with ground truth 2D detections, supervised training

BODY(3D,KPT,MVIEW,IMG)

23.1 Campus Dataset

23.1.1 Voxelpose + Voxelpose on Campus

```
@inproceedings{tumultipose,  
  title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},  
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},  
  booktitle={ECCV},  
  year={2020}  
}
```

```
@inproceedings {belagian14multi,  
  title = {{3D} Pictorial Structures for Multiple Human Pose Estimation},  
  author = {Belagiannis, Vasileios and Amin, Sikandar and Andriluka, Mykhaylo and  
↳Schiele, Bernt and Navab  
  Nassir and Ilic, Slobodan},  
  booktitle = {IEEE Computer Society Conference on Computer Vision and Pattern  
↳Recognition (CVPR)},  
  year = {2014},  
  month = {June},  
  organization={IEEE}  
}
```

Results on Campus dataset.

23.2 Panoptic Dataset

23.2.1 Voxelpose + Voxelpose + Prn64x64x64 + Cpn80x80x20 + Panoptic on Panop- tic

```
@inproceedings{tumultipose,  
  title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},  
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},  
  booktitle={ECCV},  
  year={2020}  
}
```

```
@Article = {joo_iccv_2015,  
author = {Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo_  
↪Kanade, Shohei Nobuhara, and Yaser Sheikh},  
title = {Panoptic Studio: A Massively Multiview System for Social Motion Capture},  
booktitle = {ICCV},  
year = {2015}  
}
```

Results on CMU Panoptic dataset.

23.3 Shelf Dataset

23.3.1 Voxelpose + Voxelpose on Shelf

```
@inproceedings{tumultipose,  
title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},  
author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},  
booktitle={ECCV},  
year={2020}  
}
```

```
@inproceedings {belagian14multi,  
title = {{3D} Pictorial Structures for Multiple Human Pose Estimation},  
author = {Belagiannis, Vasileios and Amin, Sikandar and Andriluka, Mykhaylo and_  
↪Schiele, Bernt and Navab  
Nassir and Ilic, Slobodan},  
booktitle = {IEEE Computer Society Conference on Computer Vision and Pattern_  
↪Recognition (CVPR)},  
year = {2014},  
month = {June},  
organization={IEEE}  
}
```

Results on Shelf dataset.

BODY(3D,MESH,SVIEW,IMG)

24.1 Mixed Dataset

24.1.1 HMR + Resnet on Mixed

```
@inProceedings{kanazawaHMR18,  
  title={End-to-end Recovery of Human Shape and Pose},  
  author = {Angjoo Kanazawa  
    and Michael J. Black  
    and David W. Jacobs  
    and Jitendra Malik},  
  booktitle={Computer Vision and Pattern Recognition (CVPR)},  
  year={2018}  
}
```

```
@inproceedings{he2016deep,  
  title={Deep residual learning for image recognition},  
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
↪recognition},  
  pages={770--778},  
  year={2016}  
}
```

```
@article{h36m_pami,  
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪  
↪Cristian},  
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing ↪  
↪in Natural Environments},  
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},  
  publisher = {IEEE Computer Society},  
  volume = {36},  
  number = {7},  
  pages = {1325-1339},  
  month = {jul},  
  year = {2014}  
}
```

Results on Human3.6M with ground-truth bounding box having MPJPE-PA of 52.60 mm on Protocol2

25.1 300w Dataset

25.1.1 Topdown Heatmap + Hrnetv2 on 300w

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},  
  journal={TPAMI},  
  year={2019}  
}
```

```
@article{sagonas2016300,  
  title={300 faces in-the-wild challenge: Database and results},  
  author={Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and  
    ↪Zafeiriou, Stefanos and Pantic, Maja},  
  journal={Image and vision computing},  
  volume={47},  
  pages={3--18},  
  year={2016},  
  publisher={Elsevier}  
}
```

Results on 300W dataset

The model is trained on 300W train.

25.2 Aflw Dataset

25.2.1 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
```

(continues on next page)

(continued from previous page)

```

    journal={TPAMI},
    year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪ facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪ workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

25.2.2 Topdown Heatmap + Hrnetv2 on Aflw

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪ facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪ workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

25.3 Coco_wholebody_face Dataset

25.3.1 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.3.2 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  (continues on next page)
```

(continued from previous page)

```
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.3.3 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}
```

```
@inproceedings{jin2020whole,
title={Whole-Body Human Pose Estimation in the Wild},
author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.3.4 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{liu2020improving,
title={Improving Convolutional Networks with Self-Calibrated Convolutions},
author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={10096--10105},
year={2020}
}
```

```
@inproceedings{jin2020whole,
title={Whole-Body Human Pose Estimation in the Wild},
author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.3.5 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.3.6 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

25.4 Cofw Dataset

25.4.1 Topdown Heatmap + Hrnetv2 on Cofw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{burgos2013robust,
  title={Robust face landmark estimation under occlusion},
  author={Burgos-Artizzu, Xavier P and Perona, Pietro and Doll{\`a}r, Piotr},
  booktitle={Proceedings of the IEEE international conference on computer vision},
  pages={1513--1520},
  year={2013}
}
```

Results on COFW dataset

The model is trained on COFW train.

25.5 WFLW Dataset

25.5.1 Deeppose + Resnet on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
```

(continues on next page)

(continued from previous page)

```

    author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, ↵
↵Qiang},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
    pages={2129--2138},
    year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

25.5.2 Deeppose + Resnet + Softwingloss on WFLW

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{lin2021structure,
  title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
  author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and ↵
↵Lu, Jiwen and Zhou, Jie},
  journal={IEEE Transactions on Image Processing},
  year={2021},
  publisher={IEEE}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, ↵
↵Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

25.5.3 Deeppose + Resnet + Wingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural
↪Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↪Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},
  year={2018},
  pages = {2235-2245},
  organization={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

25.5.4 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},
  author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
  booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
  pages={6971--6981},
  year={2019}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

25.5.5 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{wu2018look,  
  title={Look at boundary: A boundary-aware face alignment algorithm},  
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},  
  pages={2129--2138},  
  year={2018}  
}
```

Results on WFLW dataset

The model is trained on WFLW train.

25.5.6 Topdown Heatmap + Hrnetv2 on WFLW

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
          Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
          Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},  
  journal={TPAMI},  
  year={2019}  
}
```

```
@inproceedings{wu2018look,  
  title={Look at boundary: A boundary-aware face alignment algorithm},  
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},  
  pages={2129--2138},  
  year={2018}  
}
```

Results on WFLW dataset

The model is trained on WFLW train.

26.1 Deepfashion Dataset

26.1.1 Deeppose + Resnet on Deepfashion

```
@inproceedings{toshev2014deeppose,  
  title={Deeppose: Human pose estimation via deep neural networks},  
  author={Toshev, Alexander and Szegedy, Christian},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
↪recognition},  
  pages={1653--1660},  
  year={2014}  
}
```

```
@inproceedings{he2016deep,  
  title={Deep residual learning for image recognition},  
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
↪recognition},  
  pages={770--778},  
  year={2016}  
}
```

```
@inproceedings{liuQWtcvpr16DeepFashion,  
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},  
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich  
↪Annotations},  
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition  
↪(CVPR)},  
  month = {June},  
  year = {2016}  
}
```

```
@inproceedings{liuYLWteccv16FashionLandmark,  
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},  
  title = {Fashion Landmark Detection in the Wild},  
  booktitle = {European Conference on Computer Vision (ECCV)},  
  month = {October},
```

(continues on next page)

(continued from previous page)

```
year = {2016}
}
```

Results on DeepFashion val set

26.1.2 Topdown Heatmap + Resnet on Deepfashion

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuLQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWTeccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

26.2 Deepfashion2 Dataset

26.2.1 Topdown Heatmap + Resnet on Deepfashion2

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{DeepFashion2,
  author = {Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang
↪and Ping Luo},
  title={A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-
↪Identification of Clothing Images},
  journal={CVPR},
  year={2019}
}
```

Results on DeepFashion2 val set

HAND(2D,KPT,RGB,IMG)

27.1 Coco_wholebody_hand Dataset

27.1.1 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},  
  journal={TPAMI},  
  year={2019}  
}
```

```
@inproceedings{zhang2020distribution,  
  title={Distribution-aware coordinate representation for human pose estimation},  
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},  
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern  
→Recognition},  
  pages={7093--7102},  
  year={2020}  
}
```

```
@inproceedings{jin2020whole,  
  title={Whole-Body Human Pose Estimation in the Wild},  
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,  
→Chen and Ouyang, Wanli and Luo, Ping},  
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},  
  year={2020}  
}
```

Results on COCO-WholeBody-Hand val set

27.1.2 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.1.3 Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.1.4 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.1.5 Topdown Heatmap + Snet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and ↵
↵Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.1.6 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.1.7 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

27.2 Freihand2d Dataset

27.2.1 Topdown Heatmap + Resnet on Freihand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zimmermann2019freihand,
  title={Freihand: A dataset for markerless capture of hand pose and shape from single
↪rgb images},
  author={Zimmermann, Christian and Ceylan, Duygu and Yang, Jimei and Russell, Bryan and
↪Argus, Max and Brox, Thomas},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={813--822},
  year={2019}
}
```

Results on FreiHand val & test set

27.3 Interhand2d Dataset

27.3.1 Topdown Heatmap + Resnet on Interhand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
```

(continues on next page)

(continued from previous page)

```

    pages={770--778},
    year={2016}
}

```

```

@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation.
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}

```

Results on InterHand2.6M val & test set

27.4 Onehand10k Dataset

27.4.1 Deeppose + Resnet on Onehand10k

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}

```

Results on OneHand10K val set

27.4.2 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

27.4.3 Topdown Heatmap + Resnet on Onehand10k

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,  
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},  
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},  
  journal={IEEE Transactions on Circuits and Systems for Video Technology},  
  volume={29},  
  number={11},  
  pages={3258--3268},  
  year={2018},  
  publisher={IEEE}  
}
```

Results on OneHand10K val set

27.4.4 Topdown Heatmap + Hrnetv2 on Onehand10k

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},  
  journal={TPAMI},  
  year={2019}  
}
```

```
@article{wang2018mask,  
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},  
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},  
  journal={IEEE Transactions on Circuits and Systems for Video Technology},  
  volume={29},  
  number={11},  
  pages={3258--3268},  
  year={2018},  
  publisher={IEEE}  
}
```

Results on OneHand10K val set

27.4.5 Topdown Heatmap + Mobilenetv2 on Onehand10k

```
@inproceedings{sandler2018mobilenetv2,  
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},  
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and  
    ↪Chen, Liang-Chieh},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
    ↪recognition},  
  pages={4510--4520},  
  year={2018}  
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

27.4.6 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

27.5 Panoptic2d Dataset

27.5.1 Deeppose + Resnet on Panoptic2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

27.5.2 Topdown Heatmap + Mobilenetv2 on Panoptic2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},

```

(continues on next page)

(continued from previous page)

```

    pages={1145--1153},
    year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

27.5.3 Topdown Heatmap + Resnet on Panoptic2d

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

27.5.4 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},

```

(continues on next page)

(continued from previous page)

```

author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={7093--7102},
year={2020}
}

```

```

@inproceedings{simon2017hand,
title={Hand keypoint detection in single images using multiview bootstrapping},
author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
pages={1145--1153},
year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

27.5.5 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
month = {June},
year = {2020}
}

```

```

@inproceedings{simon2017hand,
title={Hand keypoint detection in single images using multiview bootstrapping},
author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
pages={1145--1153},
year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

27.5.6 Topdown Heatmap + Hrnetv2 on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
    Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

27.6 Rhd2d Dataset

27.6.1 Deeppose + Resnet on Rhd2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
}
```

(continues on next page)

(continued from previous page)

```
url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

27.6.2 Topdown Heatmap + Mobilenetv2 on Rhd2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

27.6.3 Topdown Heatmap + Resnet on Rhd2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
```

(continues on next page)

(continued from previous page)

```

title={Learning to Estimate 3D Hand Pose from Single RGB Images},
institution={arXiv:1705.01389},
year={2017},
note="https://arxiv.org/abs/1705.01389",
url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

27.6.4 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}

```

```

@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

27.6.5 Topdown Heatmap + Hrnetv2 + Dark on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

27.6.6 Topdown Heatmap + Hrnetv2 on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

HAND(3D,KPT,RGB,IMG)

28.1 Interhand3d Dataset

28.1.1 Internet + Internet on Interhand3d

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,  
  author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee, ↵  
    ↵Kyoung Mu},  
  title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation ↵  
    ↵from a Single RGB Image},  
  booktitle = {European Conference on Computer Vision (ECCV)},  
  year = {2020}  
}
```

```
@inproceedings{he2016deep,  
  title={Deep residual learning for image recognition},  
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵  
    ↵recognition},  
  pages={770--778},  
  year={2016}  
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,  
  author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee, ↵  
    ↵Kyoung Mu},  
  title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation ↵  
    ↵from a Single RGB Image},  
  booktitle = {European Conference on Computer Vision (ECCV)},  
  year = {2020}  
}
```

Results on InterHand2.6M val & test set

HAND(GESTURE,RGBD,VID)

29.1 Nvgesture Dataset

29.1.1 Mtut + I3d on Nvgesture

```
@InProceedings{Abavisani_2019_CVPR,  
  author = {Abavisani, Mahdi and Joze, Hamid Reza Vaezi and Patel, Vishal M.},  
  title = {Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With_  
↪Multimodal Training},  
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_  
↪Recognition (CVPR)},  
  month = {June},  
  year = {2019}  
}
```

```
@InProceedings{Carreira_2017_CVPR,  
  author = {Carreira, Joao and Zisserman, Andrew},  
  title = {Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset},  
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_  
↪Recognition (CVPR)},  
  month = {July},  
  year = {2017}  
}
```

```
@InProceedings{Molchanov_2016_CVPR,  
  author = {Molchanov, Pavlo and Yang, Xiaodong and Gupta, Shalini and Kim, Kihwan and_  
↪Tyree, Stephen and Kautz, Jan},  
  title = {Online Detection and Classification of Dynamic Hand Gestures With Recurrent_  
↪3D Convolutional Neural Network},  
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_  
↪Recognition (CVPR)},  
  month = {June},  
  year = {2016}  
}
```

Results on NVGesture test set

*: MTUT supports multi-modal training and uni-modal testing. Model trained with this config can be used to recognize gestures in rgb videos with [inference config](#).

WHOLEBODY

30.1 Coco-Wholebody Dataset

30.1.1 Associative Embedding + Hrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,  
  title={Associative embedding: End-to-end learning for joint detection and grouping},  
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},  
  booktitle={Advances in neural information processing systems},  
  pages={2277--2287},  
  year={2017}  
}
```

```
@inproceedings{sun2019deep,  
  title={Deep high-resolution representation learning for human pose estimation},  
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
→recognition},  
  pages={5693--5703},  
  year={2019}  
}
```

```
@inproceedings{jin2020whole,  
  title={Whole-Body Human Pose Estimation in the Wild},  
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,  
→Chen and Ouyang, Wanli and Luo, Ping},  
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},  
  year={2020}  
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

30.1.2 Associative Embedding + Higherhrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵ Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

30.1.3 Topdown Heatmap + Resnet on Coco-Wholebody

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵ Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

30.1.4 Topdown Heatmap + Hrnet on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

30.1.5 Topdown Heatmap + Tcformer on Coco-Wholebody

```
@inproceedings{zeng2022not,
  title={Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering
↪Transformer},
  author={Zeng, Wang and Jin, Sheng and Liu, Wentao and Qian, Chen and Luo, Ping and
↪Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={111101--11111},
  year={2022}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

30.1.6 Topdown Heatmap + Vipnas on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

30.1.7 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

30.1.8 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

30.2 Halpe Dataset

30.2.1 Topdown Heatmap + Hrnet + Dark on Halpe

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
```

(continues on next page)

(continued from previous page)

```
pages={7093--7102},  
year={2020}  
}
```

```
@inproceedings{li2020pastanet,  
  title={PaStaNet: Toward Human Activity Knowledge Engine},  
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and  
↪ Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},  
  booktitle={CVPR},  
  year={2020}  
}
```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

ALGORITHMS

31.1 MTUT (CVPR'2019)

31.1.1 Mtut + I3d on Nvgesture

```
@InProceedings{Abavisani_2019_CVPR,  
  author = {Abavisani, Mahdi and Joze, Hamid Reza Vaezi and Patel, Vishal M.},  
  title = {Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With_↵  
↵Multimodal Training},  
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_↵  
↵Recognition (CVPR)},  
  month = {June},  
  year = {2019}  
}
```

```
@InProceedings{Carreira_2017_CVPR,  
  author = {Carreira, Joao and Zisserman, Andrew},  
  title = {Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset},  
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_↵  
↵Recognition (CVPR)},  
  month = {July},  
  year = {2017}  
}
```

```
@InProceedings{Molchanov_2016_CVPR,  
  author = {Molchanov, Pavlo and Yang, Xiaodong and Gupta, Shalini and Kim, Kihwan and_↵  
↵Tyree, Stephen and Kautz, Jan},  
  title = {Online Detection and Classification of Dynamic Hand Gestures With Recurrent_↵  
↵3D Convolutional Neural Network},  
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_↵  
↵Recognition (CVPR)},  
  month = {June},  
  year = {2016}  
}
```

Results on NVGesture test set

*: MTUT supports multi-modal training and uni-modal testing. Model trained with this config can be used to recognize gestures in rgb videos with [inference config](#).

31.2 MSPN (ArXiv'2019)

31.2.1 Topdown Heatmap + MSPN on Coco

```
@article{li2019rethinking,
  title={Rethinking on Multi-Stage Networks for Human Pose Estimation},
  author={Li, Wenbo and Wang, Zhicheng and Yin, Binyi and Peng, Qixiang and Du, Yuming,
  ↪and Xiao, Tianzi and Yu, Gang and Lu, Hongtao and Wei, Yichen and Sun, Jian},
  journal={arXiv preprint arXiv:1901.00148},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and,
  ↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.3 InterNet (ECCV'2020)

31.3.1 Internet + Internet on Interhand3d

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
  author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
  ↪Kyoung Mu},
  title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation,
  ↪from a Single RGB Image},
  booktitle = {European Conference on Computer Vision (ECCV)},
  year = {2020}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern,
  ↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
  author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
  ↪Kyoung Mu},
```

(continues on next page)

(continued from previous page)

```

title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation.↵
↵from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}

```

Results on InterHand2.6M val & test set

31.4 DEKR (CVPR'2021)

31.4.1 Dekr + Hrnet on Coco

```

@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
↵,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern.↵
↵Recognition},
  pages={14676--14686},
  year={2021}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern.↵
↵recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and.↵
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

The results of models provided by the authors on COCO val2017 using the same evaluation protocol

The discrepancy between these results and that shown in paper is attributed to the differences in implementation details in evaluation process.

31.4.2 Dekr + Hrnet on Crowdpose

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪ Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪ Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

31.5 HigherHRNet (CVPR'2020)

31.5.1 Associative Embedding + Higherhrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
  ↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
  ↪ Thomas S and Zhang, Lei},
  ↪
```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={5386--5395},
year={2020}
}

```

```

@article{wu2017ai,
title={Ai challenger: A large-scale dataset for going deeper in image understanding},
author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
journal={arXiv preprint arXiv:1711.06475},
year={2017}
}

```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.5.2 Associative Embedding + Higherhrnet + Udp on Coco

```

@inproceedings{newell2017associative,
title={Associative embedding: End-to-end learning for joint detection and grouping},
author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
booktitle={Advances in neural information processing systems},
pages={2277--2287},
year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={5386--5395},
year={2020}
}

```

```

@InProceedings{Huang_2020_CVPR,
author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
month = {June},
year = {2020}
}

```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

31.5.3 Associative Embedding + Higherhrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.5.4 Associative Embedding + Higherhrnet on Crowdpose

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and ↵
↵ Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 2 scales ([2, 1]) are used

31.5.5 Associative Embedding + Higherhrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

(continues on next page)

(continued from previous page)

```

    year={2020}
}

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.6 DeepPose (CVPR'2014)

31.6.1 Deeppose + Resnet on Coco

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.6.2 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.6.3 Deeppose + Resnet on Mpii

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.6.4 Deeppose + Resnet + Rle on Mpii

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.6.5 Deeppose + Resnet on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, ↪
  ↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.6.6 Deeppose + Resnet + Softwingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{lin2021structure,
  title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
  author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and
↵Lu, Jiwen and Zhou, Jie},
  journal={IEEE Transactions on Image Processing},
  year={2021},
  publisher={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↵Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.6.7 Deeppose + Resnet + Wingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural
↪ Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↪ Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},
  year={2018},
  pages = {2235-2245},
  organization={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪ Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.6.8 Deeppose + Resnet on Deepfashion

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWteccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

31.6.9 Deeppose + Resnet on Onehand10k

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

31.6.10 Deeppose + Resnet on Panoptic2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

31.6.11 Deeppose + Resnet on Rhd2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.7 RLE (ICCV'2021)

31.7.1 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
}
```

(continues on next page)

(continued from previous page)

```

    year={2021}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.7.2 Deeppose + Resnet + Rle on Mpii

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},

```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

31.8 SoftWingloss (TIP'2021)

31.8.1 Deeppose + Resnet + Softwingloss on WFLW

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{lin2021structure,
  title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
  author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and
  ↪Lu, Jiwen and Zhou, Jie},
  journal={IEEE Transactions on Image Processing},
  year={2021},
  publisher={IEEE}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
  ↪Qiang},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

31.9 VideoPose3D (CVPR'2019)

31.9.1 Video Pose Lift + Videopose3d on H36m

```

@inproceedings{pavullo20193d,
  title={3d human pose estimation in video with temporal convolutions and semi-
↪supervised training},
  author={Pavullo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli,
↪Michael},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7753--7762},
  year={2019}
}

```

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu,
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}

```

Results on Human3.6M dataset with ground truth 2D detections, supervised training

Results on Human3.6M dataset with CPN 2D detections¹, supervised training

Results on Human3.6M dataset with ground truth 2D detections, semi-supervised training

Results on Human3.6M dataset with CPN 2D detections¹, semi-supervised training

¹ CPN 2D detections are provided by [official repo](#). The reformatted version used in this repository can be downloaded from [train_detection](#) and [test_detection](#).

31.9.2 Video Pose Lift + Videopose3d on Mpi_inf_3dhp

```
@inproceedings{pavullo20193d,
  title={3d human pose estimation in video with temporal convolutions and semi-
  ↳supervised training},
  author={Pavullo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli,
  ↳Michael},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↳Recognition},
  pages={7753--7762},
  year={2019}
}
```

```
@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and
  ↳Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
  ↳,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
  year = {2017},
  organization={IEEE},
  doi={10.1109/3dv.2017.00064},
}
```

Results on MPI-INF-3DHP dataset with ground truth 2D detections, supervised training

31.10 Hourglass (ECCV'2016)

31.10.1 Topdown Heatmap + Hourglass on Coco

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↳Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.10.2 Topdown Heatmap + Hourglass on Mpii

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.10.3 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↪
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

31.10.4 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

31.11 LiteHRNet (CVPR'2021)

31.11.1 Topdown Heatmap + Litehrnet on Coco

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and ↵
↵Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and ↵
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.11.2 Topdown Heatmap + Litehrnet on Mpii

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.11.3 Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

31.12 AdaptiveWingloss (ICCV'2019)

31.12.1 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},
  author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
  booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
  pages={6971--6981},
  year={2019}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
    ↪ Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪ recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.13 SimpleBaseline2D (ECCV'2018)

31.13.1 Topdown Heatmap + Resnet on Animalpose

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@InProceedings{Cao_2019_ICCV,
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,
    ↪ Cewu and Tai, Yu-Wing},
  title = {Human Pose Estimation by Convolutional Neural Networks}
```

(continues on next page)

(continued from previous page)

```

    title = {Cross-Domain Adaptation for Animal Pose Estimation},
    booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
    month = {October},
    year = {2019}
}

```

Results on AnimalPose validation set (1117 instances)

31.13.2 Topdown Heatmap + Resnet on Ap10k

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng_
↪ Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```

Results on AP-10K validation set

31.13.3 Topdown Heatmap + Resnet on Atrw

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}

```

Results on ATRW validation set

31.13.4 Topdown Heatmap + Resnet on Fly

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{pereira2019fast,
  title={Fast animal pose estimation using deep neural networks},
  author={Pereira, Talmo D and Aldarondo, Diego E and Willmore, Lindsay and Kislin,
↪Mikhail and Wang, Samuel S-H and Murthy, Mala and Shaevitz, Joshua W},
  journal={Nature methods},
  volume={16},
  number={1},
  pages={117--125},
  year={2019},
  publisher={Nature Publishing Group}
}
```

Results on Vinegar Fly test set

31.13.5 Topdown Heatmap + Resnet on Horse10

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekgonul,
↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

31.13.6 Topdown Heatmap + Resnet on Locust

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation,
  ↪ using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,
  ↪ Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

Results on Desert Locust test set

31.13.7 Topdown Heatmap + Resnet on Macaque

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel 'in the wild' macaque monkey pose dataset for markerless,
  ↪ motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,
  ↪ Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi,
  ↪ and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

31.13.8 Topdown Heatmap + Resnet on Zebra

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation,
  → using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,
  → Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

Results on Grévy's Zebra test set

31.13.9 Topdown Heatmap + Resnet on Aic

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  → recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
  → Liang, Rui and Wang, Wenjia and Zhou, Shipai and Lin, Guosen and Fu, Yanwei and others}
  → ,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC val set with ground-truth bounding boxes

31.13.10 Topdown Heatmap + Resnet + Fp16 on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.13.11 Topdown Heatmap + Resnet + Dark on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.13.12 Topdown Heatmap + Swin on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{liu2021swin,
  title={Swin transformer: Hierarchical vision transformer using shifted windows},
  author={Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang,
↪Zheng and Lin, Stephen and Guo, Baining},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={10012--10022},
  year={2021}
}
```

```
@inproceedings{lin2017feature,
  title={Feature pyramid networks for object detection},
  author={Lin, Tsung-Yi and Doll{\`a}r, Piotr and Girshick, Ross and He, Kaiming and
↪Hariharan, Bharath and Belongie, Serge},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2117--2125},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.13.13 Topdown Heatmap + Resnet on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.13.14 Topdown Heatmap + Resnet on Crowdpose

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪ Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test with YOLOv3 human detector

31.13.15 Topdown Heatmap + Resnet on JHMDB

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```



```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. *NO* test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

31.13.16 Topdown Heatmap + Resnet on MHP

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
}
```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

```

Results on MHP v2.0 val set

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. Please be cautious if you use the results in papers.

31.13.17 Topdown Heatmap + Resnet on Mpii

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

31.13.18 Topdown Heatmap + Resnet + Mpii on Mpii_trb

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{duan2019trb,
  title={TRB: A Novel Triplet Representation for Understanding 2D Human Body},
  author={Duan, Haodong and Lin, Kwan-Yee and Jin, Sheng and Liu, Wentao and Qian, Chen
↪and Ouyang, Wanli},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={9479--9488},
  year={2019}
}
```

Results on MPII-TRB val set

31.13.19 Topdown Heatmap + Resnet on Ochuman

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
  year={2019}
}
```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

31.13.20 Topdown Heatmap + Resnet on Posetrack18

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

31.13.21 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
```

(continues on next page)

(continued from previous page)

```

pages={770--778},
year={2016}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

31.13.22 Topdown Heatmap + Resnet on Deepfashion

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{liuLQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}

```

```

@inproceedings{liuYLWteccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}

```

Results on DeepFashion val set

31.13.23 Topdown Heatmap + Resnet on Deepfashion2

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{DeepFashion2,
  author = {Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang
↪and Ping Luo},
  title={A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-
↪Identification of Clothing Images},
  journal={CVPR},
  year={2019}
}
```

Results on DeepFashion2 val set

31.13.24 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},

```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Hand val set

31.13.25 Topdown Heatmap + Resnet on Freihand2d

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{zimmermann2019freihand,
  title={Freihand: A dataset for markerless capture of hand pose and shape from single ↵
↵rgb images},
  author={Zimmermann, Christian and Ceylan, Duygu and Yang, Jimei and Russell, Bryan and ↵
↵Argus, Max and Brox, Thomas},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={813--822},
  year={2019}
}

```

Results on FreiHand val & test set

31.13.26 Topdown Heatmap + Resnet on Interhand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

Results on InterHand2.6M val & test set

31.13.27 Topdown Heatmap + Resnet on Onehand10k

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
```

(continues on next page)

(continued from previous page)

```

author={Wang, Yangang and Peng, Cong and Liu, Yebin},
journal={IEEE Transactions on Circuits and Systems for Video Technology},
volume={29},
number={11},
pages={3258--3268},
year={2018},
publisher={IEEE}
}

```

Results on OneHand10K val set

31.13.28 Topdown Heatmap + Resnet on Panoptic2d

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

31.13.29 Topdown Heatmap + Resnet on Rhd2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.13.30 Topdown Heatmap + Resnet on Coco-Wholebody

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

31.14 PoseWarper (NeurIPS'2019)

31.14.1 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,
  title = {Learning Temporal Pose Estimation from Sparsely Labeled Videos},
  author = {Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo,
    ↪and Torresani, Lorenzo},
  booktitle = {Advances in Neural Information Processing Systems 33},
  year = {2019},
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
    ↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
    ↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
    ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)1.

1 Please download the precomputed human bounding boxes on PoseTrack2018 val from \$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json and place it here: \$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json to be consistent with the config. Please refer to DATA Preparation for more detail about data preparation.

31.15 SimpleBaseline3D (ICCV'2017)

31.15.1 Pose Lift + Simplebaseline3d on H36m

```
@inproceedings{martinez_2017_3dbaseline,
  title={A simple yet effective baseline for 3d human pose estimation},
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},
  booktitle={ICCV},
  year={2017}
}
```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, 
  ↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
  ↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

31.15.2 Pose Lift + Simplebaseline3d on Mpi_inf_3dhp

```
@inproceedings{martinez_2017_3dbaseline,
  title={A simple yet effective baseline for 3d human pose estimation},
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},
  booktitle={ICCV},
  year={2017}
}
```

```
@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and↪
  ↪Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
  ↪,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
```

(continues on next page)

(continued from previous page)

```
url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
year = {2017},
organization={IEEE},
doi={10.1109/3dv.2017.00064},
}
```

Results on MPI-INF-3DHP dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

31.16 HMR (CVPR'2018)

31.16.1 HMR + Resnet on Mixed

```
@inProceedings{kanazawaHMR18,
  title={End-to-end Recovery of Human Shape and Pose},
  author = {Angjoo Kanazawa
    and Michael J. Black
    and David W. Jacobs
    and Jitendra Malik},
  booktitle={Computer Vision and Pattern Recognition (CVPR)},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M with ground-truth bounding box having MPJPE-PA of 52.60 mm on Protocol2

31.17 UDP (CVPR'2020)

31.17.1 Associative Embedding + Higherhrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵ ,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↵ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

31.17.2 Associative Embedding + Hrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

31.17.3 Topdown Heatmap + Hrnet + Udp on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of DARK.

31.17.4 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
```

(continues on next page)

(continued from previous page)

```

pages={3258--3268},
year={2018},
publisher={IEEE}
}

```

Results on OneHand10K val set

31.17.5 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵ ,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↵ Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

31.17.6 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.18 ViPNAS (CVPR'2021)

31.18.1 Topdown Heatmap + Vipnas on Coco

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and_
↵ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↵ recognition},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↵ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.18.2 Topdown Heatmap + Vipnas on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

31.18.3 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

31.19 Wingloss (CVPR'2018)

31.19.1 Deeppose + Resnet + Wingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural
↪Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↪ Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},
  year={2018},
  pages = {2235-2245},
  organization={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.20 DarkPose (CVPR'2020)

31.20.1 Topdown Heatmap + Resnet + Dark on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.20.2 Topdown Heatmap + Hrnet + Dark on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.20.3 Topdown Heatmap + Hrnet + Dark on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.20.4 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern.},
  ↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for.},
  ↪facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV.},
  ↪workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}
```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

31.20.5 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

31.20.6 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```



```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.20.7 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

31.20.8 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

31.20.9 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  Recognition},
  pages={7093--7102},
  year={2020}
}
```

(continues on next page)

(continued from previous page)

```

year={2020}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

31.20.10 Topdown Heatmap + Hrnetv2 + Dark on Rhd2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

31.20.11 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

31.20.12 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.20.13 Topdown Heatmap + Hrnet + Dark on Halpe

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and
↪Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},
  year={2020}
}
```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

31.21 Associative Embedding (NIPS'2017)

31.21.1 Associative Embedding + Hrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.2 Associative Embedding + Higherhrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
```

(continues on next page)

(continued from previous page)

```

pages={5386--5395},
year={2020}
}

```

```

@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}

```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.3 Associative Embedding + Higherhrnet + Udp on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},

```

(continues on next page)

(continued from previous page)

```

author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

31.21.4 Associative Embedding + Mobilenetv2 on Coco

```

@inproceedings{newell2017associative,
title={Associative embedding: End-to-end learning for joint detection and grouping},
author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
booktitle={Advances in neural information processing systems},
pages={2277--2287},
year={2017}
}

```

```

@inproceedings{sandler2018mobilenetv2,
title={Mobilenetv2: Inverted residuals and linear bottlenecks},
author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={4510--4520},
year={2018}
}

```

```

@inproceedings{lin2014microsoft,
title={Microsoft coco: Common objects in context},
author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.5 Associative Embedding + Hrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.6 Associative Embedding + Higherhrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↵Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↵Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={5386--5395},
}
```

(continues on next page)

(continued from previous page)

```

year={2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.7 Associative Embedding + Resnet on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.8 Associative Embedding + Hrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

31.21.9 Associative Embedding + Hourglass + Ae on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.10 Associative Embedding + Higherhrnet on Crowdpose

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 2 scales ([2, 1]) are used

31.21.11 Associative Embedding + Hrnet on MHP

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}
```

Results on MHP v2.0 validation set without multi-scale test

Results on MHP v2.0 validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.21.12 Associative Embedding + Hrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.21.13 Associative Embedding + Higherhrnet on Coco-Wholebody

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.22 VoxelPose (ECCV'2020)

31.22.1 Voxelpose + Voxelpose on Campus

```
@inproceedings{tumultipose,
  title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},
  booktitle={ECCV},
  year={2020}
}
```

```
@inproceedings {belagian14multi,
  title = {{3D} Pictorial Structures for Multiple Human Pose Estimation},
  author = {Belagiannis, Vasileios and Amin, Sikandar and Andriluka, Mykhaylo and
↪Schiele, Bernt and Navab
  Nassir and Ilic, Slobodan},
  booktitle = {IEEE Computer Society Conference on Computer Vision and Pattern
↪Recognition (CVPR)},
  year = {2014},
  month = {June},
  organization={IEEE}
}
```

Results on Campus dataset.

31.22.2 Voxelpose + Voxelpose + Prn64x64x64 + Cpn80x80x20 + Panoptic on Panoptic

```
@inproceedings{tumultipose,
  title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},
  booktitle={ECCV},
  year={2020}
}
```

```
@Article = {joo_iccv_2015,
author = {Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo
↪Kanade, Shohei Nobuhara, and Yaser Sheikh},
title = {Panoptic Studio: A Massively Multiview System for Social Motion Capture},
booktitle = {ICCV},
year = {2015}
}
```

Results on CMU Panoptic dataset.

31.22.3 Voxelpose + Voxelpose on Shelf

```
@inproceedings{tumultipose,
  title={Voxelpose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},
  booktitle={ECCV},
  year={2020}
}
```

```
@inproceedings {belagian14multi,
  title = {{3D} Pictorial Structures for Multiple Human Pose Estimation},
  author = {Belagiannis, Vasileios and Amin, Sikandar and Andriluka, Mykhaylo and
↪Schiele, Bernt and Navab
  Nassir and Ilic, Slobodan},
  booktitle = {IEEE Computer Society Conference on Computer Vision and Pattern
↪Recognition (CVPR)},
  year = {2014},
  month = {June},
  organization={IEEE}
}
```

Results on Shelf dataset.

31.23 RSN (ECCV'2020)

31.23.1 Topdown Heatmap + RSN on Coco

```
@misc{cai2020learning,
  title={Learning Delicate Local Representations for Multi-Person Pose Estimation},
  author={Yuanhao Cai and Zhicheng Wang and Zhengxiong Luo and Binyi Yin and Angang Du
↪and Haoqian Wang and Xinyu Zhou and Erjin Zhou and Xiangyu Zhang and Jian Sun},
  year={2020},
  eprint={2003.04030},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.24 CID (CVPR'2022)

31.24.1 Cid + Hrnet on Coco

```
@InProceedings{Wang_2022_CVPR,
  author    = {Wang, Dongkai and Zhang, Shiliang},
  title     = {Contextual Instance Decoupling for Robust Multi-Person Pose Estimation},
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↳ Recognition (CVPR)},
  month     = {June},
  year      = {2022},
  pages     = {11060-11068}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↳ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

31.25 CPM (CVPR'2016)

31.25.1 Topdown Heatmap + CPM on Coco

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
  ↳ Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↳ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.25.2 Topdown Heatmap + CPM on JHMDB

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. NO test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

31.25.3 Topdown Heatmap + CPM on Mpii

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.26 HRNet (CVPR'2019)

31.26.1 Topdown Heatmap + Hrnet on Animalpose

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Cao_2019_ICCV,
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,
↪ Cewu and Tai, Yu-Wing},
  title = {Cross-Domain Adaptation for Animal Pose Estimation},
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
  month = {October},
  year = {2019}
}
```

Results on AnimalPose validation set (1117 instances)

31.26.2 Topdown Heatmap + Hrnet on Ap10k

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng_
↪Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

Results on AP-10K validation set

31.26.3 Topdown Heatmap + Hrnet on Atrw

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}
```

Results on ATRW validation set

31.26.4 Topdown Heatmap + Hrnet on Horse10

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekgonul,
↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

31.26.5 Topdown Heatmap + Hrnet on Macaque

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel 'in the wild'macaque monkey pose dataset for markerless
↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,
↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi
↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

31.26.6 Associative Embedding + Hrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipai and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.26.7 Topdown Heatmap + Hrnet on Aic

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC val set with ground-truth bounding boxes

31.26.8 Associative Embedding + Hrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  year={2014}
```

(continues on next page)

(continued from previous page)

```

pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.26.9 Associative Embedding + Hrnet + Udp on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

31.26.10 Dekr + Hrnet on Coco

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

The results of models provided by the authors on COCO val2017 using the same evaluation protocol

The discrepancy between these results and that shown in paper is attributed to the differences in implementation details in evaluation process.

31.26.11 Topdown Heatmap + Hrnet + Fp16 on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```



```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.26.12 Topdown Heatmap + Hrnet on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.26.13 Topdown Heatmap + Hrnet + Augmentation on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{buslaev2020albumentations,
  title={Albumentations: fast and flexible image augmentations},
  author={Buslaev, Alexander and Iglovikov, Vladimir I and Khvedchenya, Eugene and
↪Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A},
  journal={Information},
  volume={11},
  number={2},
  pages={125},
  year={2020},
  publisher={Multidisciplinary Digital Publishing Institute}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.26.14 Topdown Heatmap + Hrnet + Udp on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
```

(continues on next page)

(continued from previous page)

```

month = {June},
year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of DARK.

31.26.15 Topdown Heatmap + Hrnet + Dark on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.26.16 Dekr + Hrnet on Crowdpose

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

31.26.17 Topdown Heatmap + Hrnet on Crowdpose

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},

```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

```

Results on CrowdPose test with YOLOv3 human detector

31.26.18 Topdown Heatmap + Hrnet on H36m

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu,
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}

```

Results on Human3.6M test set with ground truth 2D detections

31.26.19 Associative Embedding + Hrnet on MHP

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},

```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}

```

Results on MHP v2.0 validation set without multi-scale test

Results on MHP v2.0 validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

31.26.20 Topdown Heatmap + Hrnet on Mpii

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

31.26.21 Topdown Heatmap + Hrnet + Dark on Mpii

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

31.26.22 Topdown Heatmap + Hrnet on Ochuman

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪ and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={889--898},
  year={2019}
}
```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

31.26.23 Topdown Heatmap + Hrnet on Posetrack18

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with MMDetection pre-trained Cascade R-CNN (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

31.26.24 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,
  title = {Learning Temporal Pose Estimation from Sparsely Labeled Videos},
  author = {Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo,
↪and Torresani, Lorenzo},
  booktitle = {Advances in Neural Information Processing Systems 33},
  year = {2019},
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
```

(continues on next page)

(continued from previous page)

```

    booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern_
↪Recognition},
    pages={5167--5176},
    year={2018}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)¹.

¹ Please download the precomputed human bounding boxes on PoseTrack2018 val from \$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json and place it here: \$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json to be consistent with the [config](#). Please refer to DATA Preparation for more detail about data preparation.

31.26.25 Associative Embedding + Hrnet on Coco-Wholebody

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.26.26 Topdown Heatmap + Hrnet on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

31.26.27 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
```

(continues on next page)

(continued from previous page)

```

pages={7093--7102},
year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

31.26.28 Topdown Heatmap + Hrnet + Dark on Halpe

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and ↵
↵Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},
  year={2020}
}

```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

31.27 HRNetv2 (TPAMI'2019)

31.27.1 Topdown Heatmap + Hrnetv2 on 300w

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{sagonas2016300,
  title={300 faces in-the-wild challenge: Database and results},
  author={Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and
    ↪Zafeiriou, Stefanos and Pantic, Maja},
  journal={Image and vision computing},
  volume={47},
  pages={3--18},
  year={2016},
  publisher={Elsevier}
}
```

Results on 300W dataset

The model is trained on 300W train.

31.27.2 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    ↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
    ↪facial landmark localization},
```

(continues on next page)

(continued from previous page)

```

author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
booktitle={2011 IEEE international conference on computer vision workshops (ICCV↵
↵workshops)},
pages={2144--2151},
year={2011},
organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

31.27.3 Topdown Heatmap + Hrnetv2 on Aflw

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@inproceedings{koestinger2011annotated,
title={Annotated facial landmarks in the wild: A large-scale, real-world database for↵
↵facial landmark localization},
author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
booktitle={2011 IEEE international conference on computer vision workshops (ICCV↵
↵workshops)},
pages={2144--2151},
year={2011},
organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

31.27.4 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

31.27.5 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

31.27.6 Topdown Heatmap + Hrnetv2 on Cofw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

@inproceedings{burgos2013robust,
  title={Robust face landmark estimation under occlusion},
  author={Burgos-Artizzu, Xavier P and Perona, Pietro and Doll{\`a}r, Piotr},
  booktitle={Proceedings of the IEEE international conference on computer vision},
  pages={1513--1520},
  year={2013}
}

```

Results on COFW dataset

The model is trained on COFW train.

31.27.7 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},
  author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
  booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
  pages={6971--6981},
  year={2019}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

31.27.8 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
    Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

31.27.9 Topdown Heatmap + Hrnetv2 on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
    Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},

```

(continues on next page)

(continued from previous page)

```

pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

31.27.10 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
    Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Hand val set

31.27.11 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```
@inproceedings{jin2020whole,  
  title={Whole-Body Human Pose Estimation in the Wild},  
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},  
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},  
  year={2020}  
}
```

Results on COCO-WholeBody-Hand val set

31.27.12 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```
@article{WangSCJDZLMTWLX19,  
  title={Deep High-Resolution Representation Learning for Visual Recognition},  
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and  
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and  
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},  
  journal={TPAMI},  
  year={2019}  
}
```

```
@inproceedings{zhang2020distribution,  
  title={Distribution-aware coordinate representation for human pose estimation},  
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},  
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},  
  pages={7093--7102},  
  year={2020}  
}
```

```
@article{wang2018mask,  
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},  
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},  
  journal={IEEE Transactions on Circuits and Systems for Video Technology},  
  volume={29},  
  number={11},  
  pages={3258--3268},  
  year={2018},  
  publisher={IEEE}  
}
```

Results on OneHand10K val set

31.27.13 Topdown Heatmap + Hrnetv2 on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

31.27.14 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
```

(continues on next page)

(continued from previous page)

```

    pages={3258--3268},
    year={2018},
    publisher={IEEE}
}

```

Results on OneHand10K val set

31.27.15 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
    Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

31.27.16 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

31.27.17 Topdown Heatmap + Hrnetv2 on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

31.27.18 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.27.19 Topdown Heatmap + Hrnetv2 + Dark on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↵ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.27.20 Topdown Heatmap + Hrnetv2 on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

31.28 SCNet (CVPR'2020)

31.28.1 Topdown Heatmap + Scnet on Coco

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
    ↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    ↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
```

(continues on next page)

(continued from previous page)

```

author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

31.28.2 Topdown Heatmap + Scnet on Mpii

```

@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}

```

```

@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

31.28.3 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face

```

@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},

```

(continues on next page)

(continued from previous page)

```
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}
```

Results on COCO-WholeBody-Face val set

31.28.4 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

BACKBONES

32.1 MSPN (ArXiv'2019)

32.1.1 Topdown Heatmap + MSPN on Coco

```
@article{li2019rethinking,  
  title={Rethinking on Multi-Stage Networks for Human Pose Estimation},  
  author={Li, Wenbo and Wang, Zhicheng and Yin, Binyi and Peng, Qixiang and Du, Yuming_↵  
↵and Xiao, Tianzi and Yu, Gang and Lu, Hongtao and Wei, Yichen and Sun, Jian},  
  journal={arXiv preprint arXiv:1901.00148},  
  year={2019}  
}
```

```
@inproceedings{lin2014microsoft,  
  title={Microsoft coco: Common objects in context},  
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_↵  
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},  
  booktitle={European conference on computer vision},  
  pages={740--755},  
  year={2014},  
  organization={Springer}  
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.2 MobilenetV2 (CVPR'2018)

32.2.1 Associative Embedding + Mobilenetv2 on Coco

```
@inproceedings{newell2017associative,  
  title={Associative embedding: End-to-end learning for joint detection and grouping},  
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},  
  booktitle={Advances in neural information processing systems},  
  pages={2277--2287},  
  year={2017}  
}
```

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.2.2 Topdown Heatmap + Mobilenetv2 on Coco

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.2.3 Topdown Heatmap + Mobilenetv2 on Mpii

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.2.4 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

32.2.5 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

32.2.6 Topdown Heatmap + Mobilenetv2 on Onehand10k

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

32.2.7 Topdown Heatmap + Mobilenetv2 on Panoptic2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

32.2.8 Topdown Heatmap + Mobilenetv2 on Rhd2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

32.3 HigherHRNet (CVPR'2020)

32.3.1 Associative Embedding + Higherhrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↪ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and_
↪ Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪ ,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.3.2 Associative Embedding + Higherhrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↪ Thomas S and Zhang, Lei},
```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={5386--5395},
year={2020}
}

```

```

@InProceedings{Huang_2020_CVPR,
author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
month = {June},
year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
title={Microsoft coco: Common objects in context},
author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

32.3.3 Associative Embedding + Higherhrnet on Coco

```

@inproceedings{newell2017associative,
title={Associative embedding: End-to-end learning for joint detection and grouping},
author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
booktitle={Advances in neural information processing systems},
pages={2277--2287},
year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={5386--5395},
year={2020}
}

```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.3.4 Associative Embedding + Higherhrnet on Crowdpose

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 2 scales ([2, 1]) are used

32.3.5 Associative Embedding + Higherhrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵ Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

32.4 ResNeSt (ArXiv'2020)

32.4.1 Topdown Heatmap + Resnest on Coco

```
@article{zhang2020resnest,
  title={ResNeSt: Split-Attention Networks},
  author={Zhang, Hang and Wu, Chongruo and Zhang, Zhongyue and Zhu, Yi and Zhang, Zhi ↵
↵ and Lin, Haibin and Sun, Yue and He, Tong and Muller, Jonas and Manmatha, R. and Li, ↵
↵ Mu and Smola, Alexander},
  journal={arXiv preprint arXiv:2004.08955},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and ↵
↵ Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  year={2014}
```

(continues on next page)

(continued from previous page)

```

booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.5 ResNext (CVPR'2017)

32.5.1 Topdown Heatmap + Resnext on Coco

```

@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{\`a}r, Piotr and Tu, Zhuowen and He, ↵
↵Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={1492--1500},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and ↵
↵Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.5.2 Topdown Heatmap + Resnext on Mpii

```

@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{\`a}r, Piotr and Tu, Zhuowen and He, ↵
↵Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵recognition},
  pages={1492--1500},
  year={2017}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↵,

```

(continues on next page)

(continued from previous page)

```

title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
year = {2014},
month = {June}
}

```

Results on MPII val set

32.6 ResNet (CVPR'2016)

32.6.1 Topdown Heatmap + Resnet on Aic

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}

```

Results on AIC val set with ground-truth bounding boxes

32.6.2 Associative Embedding + Resnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.6.3 Deeppose + Resnet on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.6.4 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.6.5 Topdown Heatmap + Resnet + Fp16 on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↵and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↵Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.6.6 Topdown Heatmap + Resnet + Dark on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```



```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.6.7 Topdown Heatmap + Resnet on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.6.8 Topdown Heatmap + Resnet on Crowdpose

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪ Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test with YOLOv3 human detector

32.6.9 Topdown Heatmap + Resnet on JHMDB

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. *NO* test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

32.6.10 Topdown Heatmap + Resnet on MHP

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
}
```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

```

Results on MHP v2.0 val set

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. Please be cautious if you use the results in papers.

32.6.11 Deeppose + Resnet on Mpii

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

32.6.12 Deeppose + Resnet + Rle on Mpii

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.6.13 Topdown Heatmap + Resnet on Mpii

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
```

(continues on next page)

(continued from previous page)

```

title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
year = {2014},
month = {June}
}

```

Results on MPII val set

32.6.14 Topdown Heatmap + Resnet + Mpii on Mpii_trb

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{duan2019trb,
  title={TRB: A Novel Triplet Representation for Understanding 2D Human Body},
  author={Duan, Haodong and Lin, Kwan-Yee and Jin, Sheng and Liu, Wentao and Qian, Chen
↪and Ouyang, Wanli},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={9479--9488},
  year={2019}
}

```

Results on MPII-TRB val set

32.6.15 Topdown Heatmap + Resnet on Ochuman

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
  year={2019}
}
```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

32.6.16 Topdown Heatmap + Resnet on Posetrack18

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

32.6.17 HMR + Resnet on Mixed

```
@inProceedings{kanazawaHMR18,
  title={End-to-end Recovery of Human Shape and Pose},
  author = {Angjoo Kanazawa
    and Michael J. Black
    and David W. Jacobs
    and Jitendra Malik},
  booktitle={Computer Vision and Pattern Recognition (CVPR)},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↵
↵Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↵
↵in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M with ground-truth bounding box having MPJPE-PA of 52.60 mm on Protocol2

32.6.18 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

32.6.19 Deeppose + Resnet on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
```

(continues on next page)

(continued from previous page)

```

    author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
    pages={2129--2138},
    year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

32.6.20 Deeppose + Resnet + Softwingloss on WFLW

```

@inproceedings{toshev2014deeppose,
    title={Deeppose: Human pose estimation via deep neural networks},
    author={Toshev, Alexander and Szegedy, Christian},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
    pages={1653--1660},
    year={2014}
}

```

```

@inproceedings{he2016deep,
    title={Deep residual learning for image recognition},
    author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
    pages={770--778},
    year={2016}
}

```

```

@article{lin2021structure,
    title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
    author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and
↪Lu, Jiwen and Zhou, Jie},
    journal={IEEE Transactions on Image Processing},
    year={2021},
    publisher={IEEE}
}

```

```

@inproceedings{wu2018look,
    title={Look at boundary: A boundary-aware face alignment algorithm},
    author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
    pages={2129--2138},
    year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

32.6.21 Deeppose + Resnet + Wingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural
↪Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↪Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},
  year={2018},
  pages = {2235-2245},
  organization={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

32.6.22 Deeppose + Resnet on Deepfashion

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWteccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

32.6.23 Topdown Heatmap + Resnet on Deepfashion

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuQWTCvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWTeccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

32.6.24 Topdown Heatmap + Resnet on Deepfashion2

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{DeepFashion2,
  author = {Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang
↪and Ping Luo},
```

(continues on next page)

(continued from previous page)

```

    title={A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-
    ↪Identification of Clothing Images},
    journal={CVPR},
    year={2019}
}

```

Results on DeepFashion2 val set

32.6.25 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand

```

@inproceedings{xiao2018simple,
    title={Simple baselines for human pose estimation and tracking},
    author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
    booktitle={Proceedings of the European conference on computer vision (ECCV)},
    pages={466--481},
    year={2018}
}

```

```

@inproceedings{he2016deep,
    title={Deep residual learning for image recognition},
    author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
    booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪recognition},
    pages={770--778},
    year={2016}
}

```

```

@inproceedings{jin2020whole,
    title={Whole-Body Human Pose Estimation in the Wild},
    author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
    ↪Chen and Ouyang, Wanli and Luo, Ping},
    booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
    year={2020}
}

```

Results on COCO-WholeBody-Hand val set

32.6.26 Topdown Heatmap + Resnet on Freihand2d

```

@inproceedings{xiao2018simple,
    title={Simple baselines for human pose estimation and tracking},
    author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
    booktitle={Proceedings of the European conference on computer vision (ECCV)},
    pages={466--481},
    year={2018}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zimmermann2019freihand,
  title={Freihand: A dataset for markerless capture of hand pose and shape from single_
↪rgb images},
  author={Zimmermann, Christian and Ceylan, Duygu and Yang, Jimei and Russell, Bryan and_
↪Argus, Max and Brox, Thomas},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={813--822},
  year={2019}
}
```

Results on FreiHand val & test set

32.6.27 Topdown Heatmap + Resnet on Interhand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,_
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation_
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

Results on InterHand2.6M val & test set

32.6.28 Deeppose + Resnet on Onehand10k

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

32.6.29 Topdown Heatmap + Resnet on Onehand10k

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```



```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

32.6.30 Deeppose + Resnet on Panoptic2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

32.6.31 Topdown Heatmap + Resnet on Panoptic2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↵Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

32.6.32 Deeppose + Resnet on Rhd2d

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

32.6.33 Topdown Heatmap + Resnet on Rhd2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={770--778},
  year={2016}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

32.6.34 Internet + Internet on Interhand3d

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
  author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪ Kyoung Mu},
  title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation
↪ from a Single RGB Image},
  booktitle = {European Conference on Computer Vision (ECCV)},
```

(continues on next page)

(continued from previous page)

```
year = {2020}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

Results on InterHand2.6M val & test set

32.7 Hourglass (ECCV'2016)

32.7.1 Topdown Heatmap + Hourglass on Coco

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.7.2 Topdown Heatmap + Hourglass on Mpii

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.7.3 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↪
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

32.7.4 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

32.8 ShufflenetV1 (CVPR'2018)

32.8.1 Topdown Heatmap + Shufflenetv1 on Coco

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.8.2 Topdown Heatmap + Shufflenetv1 on Mpii

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile_},
  ↪devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_},
  ↪recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.9 Swin (ICCV'2021)

32.9.1 Topdown Heatmap + Swin on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{liu2021swin,
  title={Swin transformer: Hierarchical vision transformer using shifted windows},
  author={Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang,}
  ↪Zheng and Lin, Stephen and Guo, Baining},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={10012--10022},
  year={2021}
}
```

```
@inproceedings{lin2017feature,
  title={Feature pyramid networks for object detection},
  author={Lin, Tsung-Yi and Doll{\`a}r, Piotr and Girshick, Ross and He, Kaiming and}
  ↪Hariharan, Bharath and Belongie, Serge},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_},
  ↪recognition},
```

(continues on next page)

(continued from previous page)

```

    pages={2117--2125},
    year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.10 LiteHRNet (CVPR'2021)

32.10.1 Topdown Heatmap + Litehrnet on Coco

```

@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.10.2 Topdown Heatmap + Litehrnet on Mpii

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.10.3 Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

32.11 I3D (CVPR'2017)

32.11.1 Mtut + I3d on Nvgesture

```
@InProceedings{Abavisani_2019_CVPR,
  author = {Abavisani, Mahdi and Joze, Hamid Reza Vaezi and Patel, Vishal M.},
  title = {Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With_
↪Multimodal Training},
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪Recognition (CVPR)},
  month = {June},
  year = {2019}
}
```

```
@InProceedings{Carreira_2017_CVPR,
  author = {Carreira, Joao and Zisserman, Andrew},
  title = {Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset},
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_
↪Recognition (CVPR)},
  month = {July},
  year = {2017}
}
```

```
@InProceedings{Molchanov_2016_CVPR,
  author = {Molchanov, Pavlo and Yang, Xiaodong and Gupta, Shalini and Kim, Kihwan and_
↪Tyree, Stephen and Kautz, Jan},
  title = {Online Detection and Classification of Dynamic Hand Gestures With Recurrent_
↪3D Convolutional Neural Network},
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern_
↪Recognition (CVPR)},
  month = {June},
  year = {2016}
}
```

Results on NVGesture test set

*: MTUT supports multi-modal training and uni-modal testing. Model trained with this config can be used to recognize gestures in rgb videos with [inference config](#).

32.12 ShufflenetV2 (ECCV'2018)

32.12.1 Topdown Heatmap + Shufflenetv2 on Coco

```
@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.12.2 Topdown Heatmap + Shufflenetv2 on Mpii

```
@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪ ,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.13 TCFormer (CVPR'2022)

32.13.1 Topdown Heatmap + Tcformer on Coco-Wholebody

```
@inproceedings{zeng2022not,
  title={Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering
↪ Transformer},
  author={Zeng, Wang and Jin, Sheng and Liu, Wentao and Qian, Chen and Luo, Ping and
↪ Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={11101--11111},
  year={2022}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

32.14 ResNetV1D (CVPR'2019)

32.14.1 Topdown Heatmap + Resnetv1d on Coco

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan,
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.14.2 Topdown Heatmap + Resnetv1d on Mpii

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan,
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.15 SEResNet (CVPR'2018)

32.15.1 Topdown Heatmap + Seresnet on Coco

```
@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={7132--7141},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that * means without imagenet pre-training.

32.15.2 Topdown Heatmap + Seresnet on Mpii

```
@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={7132--7141},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,  
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}  
↪ ,  
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},  
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},  
  year = {2014},  
  month = {June}  
}
```

Results on MPII val set

Note that * means without imagenet pre-training.

32.16 ViPNAS (CVPR'2021)

32.16.1 Topdown Heatmap + Vipnas on Coco

```
@article{xu2021vipnas,  
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},  
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and  
↪ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
↪ recognition},  
  year={2021}  
}
```

```
@inproceedings{lin2014microsoft,  
  title={Microsoft coco: Common objects in context},  
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and  
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},  
  booktitle={European conference on computer vision},  
  pages={740--755},  
  year={2014},  
  organization={Springer}  
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.16.2 Topdown Heatmap + Vipnas on Coco-Wholebody

```
@article{xu2021vipnas,  
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},  
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and  
↪ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},  
  booktitle={Proceedings of the IEEE conference on computer vision and pattern  
↪ recognition},  
  year={2021}  
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

32.16.3 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
  ↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

32.17 PVTv2 (CVMJ'2022)

32.17.1 Topdown Heatmap + PVT on Coco

```
@inproceedings{wang2021pyramid,
  title={Pyramid vision transformer: A versatile backbone for dense prediction without
  ↪convolutions},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao
  ↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
pages={568--578},
year={2021}
}

```

```

@article{wang2022pvt,
  title={PVT v2: Improved baselines with Pyramid Vision Transformer},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao_
↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  journal={Computational Visual Media},
  pages={1--10},
  year={2022},
  publisher={Springer}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.18 PVT (ICCV'2021)

32.18.1 Topdown Heatmap + PVT on Coco

```

@inproceedings{wang2021pyramid,
  title={Pyramid vision transformer: A versatile backbone for dense prediction without_
↪convolutions},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao_
↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={568--578},
  year={2021}
}

```

```

@article{wang2022pvt,
  title={PVT v2: Improved baselines with Pyramid Vision Transformer},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao_
↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  journal={Computational Visual Media},
  pages={1--10},
  year={2022},

```

(continues on next page)

(continued from previous page)

```

publisher={Springer}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.19 RSN (ECCV'2020)

32.19.1 Topdown Heatmap + RSN on Coco

```

@misc{cai2020learning,
  title={Learning Delicate Local Representations for Multi-Person Pose Estimation},
  author={Yuanhao Cai and Zhicheng Wang and Zhengxiong Luo and Binyi Yin and Angang Du
↪and Haoqian Wang and Xinyu Zhou and Erjin Zhou and Xiangyu Zhang and Jian Sun},
  year={2020},
  eprint={2003.04030},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.20 CPM (CVPR'2016)

32.20.1 Topdown Heatmap + CPM on Coco

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.20.2 Topdown Heatmap + CPM on JHMDB

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. NO test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

32.20.3 Topdown Heatmap + CPM on Mpii

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪ Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪ ,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.21 HRNet (CVPR'2019)

32.21.1 Topdown Heatmap + Hrnet on Animalpose

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Cao_2019_ICCV,
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,
↪ Cewu and Tai, Yu-Wing},
  title = {Cross-Domain Adaptation for Animal Pose Estimation},
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
  month = {October},
  year = {2019}
}
```

Results on AnimalPose validation set (1117 instances)

32.21.2 Topdown Heatmap + Hrnet on Ap10k

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng
↪Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

Results on AP-10K validation set

32.21.3 Topdown Heatmap + Hrnet on Atrw

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}
```

Results on ATRW validation set

32.21.4 Topdown Heatmap + Hrnet on Horse10

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekogonul,
↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

32.21.5 Topdown Heatmap + Hrnet on Macaque

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel ‘in the wild’macaque monkey pose dataset for markerless
↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,
↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi
↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

32.21.6 Associative Embedding + Hrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↵Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↵,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.21.7 Topdown Heatmap + Hrnet on Aic

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↵Liang, Rui and Wang, Wenjia and Zhou, Shiwei and Lin, Guosen and Fu, Yanwei and others}
↵,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC val set with ground-truth bounding boxes

32.21.8 Associative Embedding + Hrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.21.9 Associative Embedding + Hrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

32.21.10 Dekr + Hrnet on Coco

```

@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪Recognition},
  pages={14676--14686},
  year={2021}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},

```

(continues on next page)

(continued from previous page)

```

booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

The results of models provided by the authors on COCO val2017 using the same evaluation protocol

The discrepancy between these results and that shown in paper is attributed to the differences in implementation details in evaluation process.

32.21.11 Topdown Heatmap + Hrnet + Fp16 on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↵and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↵Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a'}, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.21.12 Topdown Heatmap + Hrnet on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.21.13 Topdown Heatmap + Hrnet + Augmentation on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{buslaev2020albumentations,
  title={Albumentations: fast and flexible image augmentations},
  author={Buslaev, Alexander and Iglovikov, Vladimir I and Khvedchenya, Eugene and
↪Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A},
  journal={Information},
  volume={11},
  number={2},
  pages={125},
  year={2020},
  publisher={Multidisciplinary Digital Publishing Institute}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},

```

(continues on next page)

(continued from previous page)

```

pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.21.14 Topdown Heatmap + Hrnet + Udp on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of [DARK](#).

32.21.15 Topdown Heatmap + Hrnet + Dark on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.21.16 Dekr + Hrnet on Crowdpose

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}

```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

32.21.17 Topdown Heatmap + Hrnet on Crowdpose

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}

```

Results on CrowdPose test with YOLOv3 human detector

32.21.18 Topdown Heatmap + Hrnet on H36m

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, 
    ↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing_
    ↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M test set with ground truth 2D detections

32.21.19 Associative Embedding + Hrnet on MHP

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
    ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a_
    ↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng_
    ↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}
```

Results on MHP v2.0 validation set without multi-scale test

Results on MHP v2.0 validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

32.21.20 Topdown Heatmap + Hrnet on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.21.21 Topdown Heatmap + Hrnet + Dark on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.21.22 Topdown Heatmap + Hrnet on Ochuman

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
  year={2019}
}
```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

32.21.23 Topdown Heatmap + Hrnet on Posetrack18

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

32.21.24 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,
  title = {Learning Temporal Pose Estimation from Sparsely Labeled Videos},
  author = {Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo,
    ↪and Torresani, Lorenzo},
  booktitle = {Advances in Neural Information Processing Systems 33},
  year = {2019},
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
    ↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
    ↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
    ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)¹.

¹ Please download the precomputed human bounding boxes on PoseTrack2018 val from \$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json and place it here: \$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json to be consistent with the [config](#). Please refer to DATA

Preparation for more detail about data preparation.

32.21.25 Associative Embedding + Hrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

32.21.26 Topdown Heatmap + Hrnet on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},

```

(continues on next page)

(continued from previous page)

```

    year={2020}
}

```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

32.21.27 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

32.21.28 Topdown Heatmap + Hrnet + Dark on Halpe

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and
↪Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},
  year={2020}
}
```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

32.22 HRNetv2 (TPAMI'2019)

32.22.1 Topdown Heatmap + Hrnetv2 on 300w

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{sagonas2016300,
  title={300 faces in-the-wild challenge: Database and results},
  author={Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and
↪Zafeiriou, Stefanos and Pantic, Maja},
  journal={Image and vision computing},
  volume={47},
  pages={3--18},
  year={2016},
  publisher={Elsevier}
}
```

Results on 300W dataset

The model is trained on 300W train.

32.22.2 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
    facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
    workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}
```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

32.22.3 Topdown Heatmap + Hrnetv2 on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
    facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
    workshops)},

```

(continues on next page)

(continued from previous page)

```

pages={2144--2151},
year={2011},
organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

32.22.4 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
    Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

32.22.5 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
}

```

(continues on next page)

(continued from previous page)

```

year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

32.22.6 Topdown Heatmap + Hrnetv2 on Cofw

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{burgos2013robust,
  title={Robust face landmark estimation under occlusion},
  author={Burgos-Artizzu, Xavier P and Perona, Pietro and Doll{'a}r, Piotr},
  booktitle={Proceedings of the IEEE international conference on computer vision},
  pages={1513--1520},
  year={2013}
}

```

Results on COFW dataset

The model is trained on COFW train.

32.22.7 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},

```

(continues on next page)

(continued from previous page)

```

author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
pages={6971--6981},
year={2019}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

32.22.8 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

32.22.9 Topdown Heatmap + Hrnetv2 on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
    Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

32.22.10 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
```

(continues on next page)

(continued from previous page)

```

author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}

```

Results on COCO-WholeBody-Hand val set

32.22.11 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@inproceedings{jin2020whole,
title={Whole-Body Human Pose Estimation in the Wild},
author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
year={2020}
}

```

Results on COCO-WholeBody-Hand val set

32.22.12 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@inproceedings{zhang2020distribution,
title={Distribution-aware coordinate representation for human pose estimation},
author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={7093--7102},
year={2020}
}

```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

32.22.13 Topdown Heatmap + Hrnetv2 on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

32.22.14 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
```

(continues on next page)

(continued from previous page)

```

    title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human↵
↵Pose Estimation},
    booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}↵
↵,
    month = {June},
    year = {2020}
}

```

```

@article{wang2018mask,
    title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
    author={Wang, Yangang and Peng, Cong and Liu, Yebin},
    journal={IEEE Transactions on Circuits and Systems for Video Technology},
    volume={29},
    number={11},
    pages={3258--3268},
    year={2018},
    publisher={IEEE}
}

```

Results on OneHand10K val set

32.22.15 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
    title={Deep High-Resolution Representation Learning for Visual Recognition},
    author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
    journal={TPAMI},
    year={2019}
}

```

```

@inproceedings{zhang2020distribution,
    title={Distribution-aware coordinate representation for human pose estimation},
    author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
    booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern↵
↵Recognition},
    pages={7093--7102},
    year={2020}
}

```

```

@inproceedings{simon2017hand,
    title={Hand keypoint detection in single images using multiview bootstrapping},
    author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
    booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern↵
↵Recognition},
    pages={1145--1153},
    year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

32.22.16 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↪ Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

32.22.17 Topdown Heatmap + Hrnetv2 on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↪ Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

32.22.18 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

32.22.19 Topdown Heatmap + Hrnetv2 + Dark on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↵ Recognition},
```

(continues on next page)

(continued from previous page)

```

pages={7093--7102},
year={2020}
}

```

```

@TechReport{zb2017hand,
author={Christian Zimmermann and Thomas Brox},
title={Learning to Estimate 3D Hand Pose from Single RGB Images},
institution={arXiv:1705.01389},
year={2017},
note="https://arxiv.org/abs/1705.01389",
url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

32.22.20 Topdown Heatmap + Hrnetv2 on Rhd2d

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@TechReport{zb2017hand,
author={Christian Zimmermann and Thomas Brox},
title={Learning to Estimate 3D Hand Pose from Single RGB Images},
institution={arXiv:1705.01389},
year={2017},
note="https://arxiv.org/abs/1705.01389",
url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

32.23 HRFormer (NIPS'2021)

32.23.1 Topdown Heatmap + Hrformer on Coco

```

@article{yuan2021hrformer,
title={HRFormer: High-Resolution Vision Transformer for Dense Predict},
author={Yuan, Yuhui and Fu, Rao and Huang, Lang and Lin, Weihong and Zhang, Chao and
        Chen, Xilin and Wang, Jingdong},
journal={Advances in Neural Information Processing Systems},
volume={34},

```

(continues on next page)

(continued from previous page)

```

    year={2021}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.24 AlexNet (NeurIPS'2012)

32.24.1 Topdown Heatmap + Alexnet on Coco

```

@inproceedings{krizhevsky2012imagenet,
  title={Imagenet classification with deep convolutional neural networks},
  author={Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E},
  booktitle={Advances in neural information processing systems},
  pages={1097--1105},
  year={2012}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.25 VGG (ICLR'2015)

32.25.1 Topdown Heatmap + VGG on Coco

```

@article{simonyan2014very,
  title={Very deep convolutional networks for large-scale image recognition},
  author={Simonyan, Karen and Zisserman, Andrew},
  journal={arXiv preprint arXiv:1409.1556},

```

(continues on next page)

(continued from previous page)

```

    year={2014}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.26 SCNet (CVPR'2020)

32.26.1 Topdown Heatmap + Snet on Coco

```

@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

32.26.2 Topdown Heatmap + Scnet on Mpii

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

32.26.3 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

32.26.4 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.1 InterHand2.6M (ECCV'2020)

33.1.1 Topdown Heatmap + Resnet on Interhand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,↪
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation↪
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

Results on InterHand2.6M val & test set

33.1.2 Internet + Internet on Interhand3d

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation.
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

```
@inproceedings{he2016deep,
title={Deep residual learning for image recognition},
author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={770--778},
year={2016}
}
```

```
@InProceedings{Moon_2020_ECCV_InterHand2.6M,
author = {Moon, Gyeongsik and Yu, Shoou-I and Wen, He and Shiratori, Takaaki and Lee,
↪Kyoung Mu},
title = {InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation.
↪from a Single RGB Image},
booktitle = {European Conference on Computer Vision (ECCV)},
year = {2020}
}
```

Results on InterHand2.6M val & test set

33.2 MPII (CVPR'2014)

33.2.1 Deeppose + Resnet on Mpii

```
@inproceedings{toshev2014deeppose,
title={Deeppose: Human pose estimation via deep neural networks},
author={Toshev, Alexander and Szegedy, Christian},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={1653--1660},
year={2014}
}
```

```
@inproceedings{he2016deep,
title={Deep residual learning for image recognition},
author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={770--778},
year={2016}
}
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}

```

Results on MPII val set

33.2.2 Deeppose + Resnet + Rle on Mpii

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},

```

(continues on next page)

(continued from previous page)

```
month = {June}
}
```

Results on MPII val set

33.2.3 Topdown Heatmap + Hrnet on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.4 Topdown Heatmap + Mobilenetv2 on Mpii

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.5 Topdown Heatmap + Resnet on Mpii

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.6 Topdown Heatmap + CPM on Mpii

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.7 Topdown Heatmap + Shufflenetv2 on Mpii

```
@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.8 Topdown Heatmap + Litehrnet on Mpii

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.9 Topdown Heatmap + Resnext on Mpii

```
@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{\`a}r, Piotr and Tu, Zhuowen and He,
↪Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1492--1500},
  year={2017}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.10 Topdown Heatmap + Shufflenetv1 on Mpii

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile
↪devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{andriluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.11 Topdown Heatmap + Hrnet + Dark on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.12 Topdown Heatmap + Scnet on Mpii

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.13 Topdown Heatmap + Seresnet on Mpii

```
@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={7132--7141},
  year={2018}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

Note that * means without imagenet pre-training.

33.2.14 Topdown Heatmap + Resnetv1d on Mpii

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.2.15 Topdown Heatmap + Hourglass on Mpii

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt},
  ↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

33.3 NVGesture (CVPR'2016)

33.3.1 Mtut + I3d on Nvgesture

```
@InProceedings{Abavisani_2019_CVPR,
  author = {Abavisani, Mahdi and Joze, Hamid Reza Vaezi and Patel, Vishal M.},
  title = {Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With ↪
  ↪Multimodal Training},
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↪
  ↪Recognition (CVPR)},
  month = {June},
  year = {2019}
}
```

```
@InProceedings{Carreira_2017_CVPR,
  author = {Carreira, Joao and Zisserman, Andrew},
  title = {Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset},
  booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern ↪
  ↪Recognition (CVPR)},
  month = {July},
  year = {2017}
}
```

```
@InProceedings{Molchanov_2016_CVPR,
  author = {Molchanov, Pavlo and Yang, Xiaodong and Gupta, Shalini and Kim, Kihwan and ↪
  ↪Tyree, Stephen and Kautz, Jan},
  title = {Online Detection and Classification of Dynamic Hand Gestures With Recurrent ↪
  ↪3D Convolutional Neural Network},
```

(continues on next page)

(continued from previous page)

```

booktitle = {Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition (CVPR)},
month = {June},
year = {2016}
}

```

Results on NVGesture test set

*: MTUT supports multi-modal training and uni-modal testing. Model trained with this config can be used to recognize gestures in rgb videos with [inference config](#).

33.4 MacaquePose (bioRxiv'2020)

33.4.1 Topdown Heatmap + Resnet on Macaque

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@article{labuguen2020macaquepose,
  title={MacaquePose: A novel 'in the wild'macaque monkey pose dataset for markerless
↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,
↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi
↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}

```

Results on MacaquePose with ground-truth detection bounding boxes

33.4.2 Topdown Heatmap + Hrnet on Macaque

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@article{labuguen2020macaquepose,
  title={MacaquePose: A novel ‘in the wild’macaque monkey pose dataset for markerless,
  ↪motion capture},
  author={Labuguen, Rollyn and Matsumoto, Jumpei and Negrete, Salvador and Nishimaru,
  ↪Hiroshi and Nishijo, Hisao and Takada, Masahiko and Go, Yasuhiro and Inoue, Ken-ichi,
  ↪and Shibata, Tomohiro},
  journal={bioRxiv},
  year={2020},
  publisher={Cold Spring Harbor Laboratory}
}
```

Results on MacaquePose with ground-truth detection bounding boxes

33.5 ATRW (ACM MM’2020)

33.5.1 Topdown Heatmap + Hrnet on Atrw

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern,
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
  booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
  pages={2590--2598},
  year={2020}
}
```

Results on ATRW validation set

33.5.2 Topdown Heatmap + Resnet on Atrw

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{li2020atrw,
  title={ATRW: A Benchmark for Amur Tiger Re-identification in the Wild},
  author={Li, Shuyuan and Li, Jianguo and Tang, Hanlin and Qian, Rui and Lin, Weiyao},
```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the 28th ACM International Conference on Multimedia},
pages={2590--2598},
year={2020}
}

```

Results on ATRW validation set

33.6 Animal-Pose (ICCV'2019)

33.6.1 Topdown Heatmap + Hrnet on Animalpose

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@InProceedings{Cao_2019_ICCV,
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,
↪ Cewu and Tai, Yu-Wing},
  title = {Cross-Domain Adaptation for Animal Pose Estimation},
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
  month = {October},
  year = {2019}
}

```

Results on AnimalPose validation set (1117 instances)

33.6.2 Topdown Heatmap + Resnet on Animalpose

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@InProceedings{Cao_2019_ICCV,
  author = {Cao, Jinkun and Tang, Hongyang and Fang, Hao-Shu and Shen, Xiaoyong and Lu,
↪ Cewu and Tai, Yu-Wing},
  title = {Cross-Domain Adaptation for Animal Pose Estimation},
  booktitle = {The IEEE International Conference on Computer Vision (ICCV)},
  month = {October},

```

(continues on next page)

(continued from previous page)

```

    year = {2019}
}

```

Results on AnimalPose validation set (1117 instances)

33.7 Human3.6M (TPAMI'2014)

33.7.1 Topdown Heatmap + Hrnet on H36m

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}

```

Results on Human3.6M test set with ground truth 2D detections

33.7.2 Pose Lift + Simplebaseline3d on H36m

```

@inproceedings{martinez_2017_3dbaseline,
  title={A simple yet effective baseline for 3d human pose estimation},
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},
  booktitle={ICCV},
  year={2017}
}

```

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
↪in Natural Environments},

```

(continues on next page)

(continued from previous page)

```

journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
publisher = {IEEE Computer Society},
volume = {36},
number = {7},
pages = {1325-1339},
month = {jul},
year = {2014}
}

```

Results on Human3.6M dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

33.7.3 Video Pose Lift + Videopose3d on H36m

```

@inproceedings{pav1lo20193d,
  title={3d human pose estimation in video with temporal convolutions and semi-
↪supervised training},
  author={Pav1lo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli, ↪
↪Michael},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↪
↪Recognition},
  pages={7753--7762},
  year={2019}
}

```

```

@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing ↪
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}

```

Results on Human3.6M dataset with ground truth 2D detections, supervised training

Results on Human3.6M dataset with CPN 2D detections¹, supervised training

Results on Human3.6M dataset with ground truth 2D detections, semi-supervised training

Results on Human3.6M dataset with CPN 2D detections¹, semi-supervised training

1 CPN 2D detections are provided by [official repo](#). The reformatted version used in this repository can be downloaded from [train_detection](#) and [test_detection](#).

33.7.4 HMR + Resnet on Mixed

```
@inproceedings{kanazawaHMR18,
  title={End-to-end Recovery of Human Shape and Pose},
  author = {Angjoo Kanazawa
    and Michael J. Black
    and David W. Jacobs
    and Jitendra Malik},
  booktitle={Computer Vision and Pattern Recognition (CVPR)},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{h36m_pami,
  author = {Ionescu, Catalin and Papava, Dragos and Olaru, Vlad and Sminchisescu, ↪
↪Cristian},
  title = {Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing↪
↪in Natural Environments},
  journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
  publisher = {IEEE Computer Society},
  volume = {36},
  number = {7},
  pages = {1325-1339},
  month = {jul},
  year = {2014}
}
```

Results on Human3.6M with ground-truth bounding box having MPJPE-PA of 52.60 mm on Protocol2

33.8 OneHand10K (TCSVT'2019)

33.8.1 Deeppose + Resnet on Onehand10k

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

33.8.2 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
}
```

(continues on next page)

(continued from previous page)

```
publisher={IEEE}
}
```

Results on OneHand10K val set

33.8.3 Topdown Heatmap + Resnet on Onehand10k

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

33.8.4 Topdown Heatmap + Hrnetv2 on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

33.8.5 Topdown Heatmap + Mobilenetv2 on Onehand10k

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

33.8.6 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
↵ booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵ ,
  month = {June},
  year = {2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}
```

Results on OneHand10K val set

33.9 JHMDB (ICCV'2013)

33.9.1 Topdown Heatmap + CPM on JHMDB

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↵ Recognition},
↵ pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. NO test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

33.9.2 Topdown Heatmap + Resnet on JHMDB

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{Jhuang:ICCV:2013,
  title = {Towards understanding action recognition},
  author = {H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black},
  booktitle = {International Conf. on Computer Vision (ICCV)},
  month = Dec,
  pages = {3192-3199},
  year = {2013}
}
```

Results on Sub-JHMDB dataset

The models are pre-trained on MPII dataset only. *NO* test-time augmentation (multi-scale /rotation testing) is used.

- Normalized by Person Size
- Normalized by Torso Size

33.10 CMU Panoptic (ICCV'2015)

33.10.1 Voxelpose + Voxelpose + Prn64x64x64 + Cpn80x80x20 + Panoptic on Panoptic

```
@inproceedings{tumultipose,
  title={VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment},
  author={Tu, Hanyue and Wang, Chunyu and Zeng, Wenjun},
  booktitle={ECCV},
  year={2020}
}
```

```
@Article = {joo_iccv_2015,
author = {Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo
↪Kanade, Shohei Nobuhara, and Yaser Sheikh},
```

(continues on next page)

(continued from previous page)

```

title = {Panoptic Studio: A Massively Multiview System for Social Motion Capture},
booktitle = {ICCV},
year = {2015}
}

```

Results on CMU Panoptic dataset.

33.11 CMU Panoptic HandDB (CVPR'2017)

33.11.1 Deeppose + Resnet on Panoptic2d

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

33.11.2 Topdown Heatmap + Mobilenetv2 on Panoptic2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

33.11.3 Topdown Heatmap + Resnet on Panoptic2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

33.11.4 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
  Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

33.11.5 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
  Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
  ,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

33.11.6 Topdown Heatmap + Hrnetv2 on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

33.12 AFLW (ICCVW'2011)

33.12.1 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  ↪
}
```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={7093--7102},
year={2020}
}

```

```

@inproceedings{koestinger2011annotated,
title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪facial landmark localization},
author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪workshops)},
pages={2144--2151},
year={2011},
organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

33.12.2 Topdown Heatmap + Hrnetv2 on Aflw

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@inproceedings{koestinger2011annotated,
title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪facial landmark localization},
author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪workshops)},
pages={2144--2151},
year={2011},
organization={IEEE}
}

```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

33.13 300W (IMAVIS'2016)

33.13.1 Topdown Heatmap + Hrnetv2 on 300w

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@article{sagonas2016300,
  title={300 faces in-the-wild challenge: Database and results},
  author={Sagonas, Christos and Antonakos, Epameinondas and Tzimiropoulos, Georgios and
    ↪Zafeiriou, Stefanos and Pantic, Maja},
  journal={Image and vision computing},
  volume={47},
  pages={3--18},
  year={2016},
  publisher={Elsevier}
}
```

Results on 300W dataset

The model is trained on 300W train.

33.14 FreiHand (ICCV'2019)

33.14.1 Topdown Heatmap + Resnet on Freihand2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zimmermann2019freihand,
  title={Freihand: A dataset for markerless capture of hand pose and shape from single_
↪rgb images},
  author={Zimmermann, Christian and Ceylan, Duygu and Yang, Jimei and Russell, Bryan and_
↪Argus, Max and Brox, Thomas},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={813--822},
  year={2019}
}
```

Results on FreiHand val & test set

33.15 MHP (ACM MM'2018)

33.15.1 Associative Embedding + Hrnet on MHP

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a_
↪new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng_
↪and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}
```

Results on MHP v2.0 validation set without multi-scale test

Results on MHP v2.0 validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.15.2 Topdown Heatmap + Resnet on MHP

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhao2018understanding,
  title={Understanding humans in crowded scenes: Deep nested adversarial learning and a
↵new benchmark for multi-human parsing},
  author={Zhao, Jian and Li, Jianshu and Cheng, Yu and Sim, Terence and Yan, Shuicheng
↵and Feng, Jiashi},
  booktitle={Proceedings of the 26th ACM international conference on Multimedia},
  pages={792--800},
  year={2018}
}
```

Results on MHP v2.0 val set

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. Please be cautious if you use the results in papers.

33.16 Vinegar Fly (Nature Methods'2019)

33.16.1 Topdown Heatmap + Resnet on Fly

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{pereira2019fast,
  title={Fast animal pose estimation using deep neural networks},
  author={Pereira, Talmo D and Aldarondo, Diego E and Willmore, Lindsay and Kislin,
↵Mikhail and Wang, Samuel S-H and Murthy, Mala and Shaevitz, Joshua W},
```

(continues on next page)

(continued from previous page)

```

journal={Nature methods},
volume={16},
number={1},
pages={117--125},
year={2019},
publisher={Nature Publishing Group}
}

```

Results on Vinegar Fly test set

33.17 CrowdPose (CVPR'2019)

33.17.1 Associative Embedding + Higherhrnet on Crowdpose

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↪ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪ Recognition},
  pages={5386--5395},
  year={2020}
}

```

```

@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and_
↪ Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}

```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 2 scales ([2, 1]) are used

33.17.2 Dekr + Hrnet on Crowdpose

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
  ↪,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}
```

Results on CrowdPose test without multi-scale test

Results on CrowdPose test with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

33.17.3 Topdown Heatmap + Hrnet on Crowdpose

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
  ↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},

```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

```

Results on CrowdPose test with YOLOv3 human detector

33.17.4 Topdown Heatmap + Resnet on Crowdpose

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{li2018crowdpose,
  title={CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark},
  author={Li, Jiefeng and Wang, Can and Zhu, Hao and Mao, Yihuan and Fang, Hao-Shu and
↪Lu, Cewu},
  journal={arXiv preprint arXiv:1812.00324},
  year={2018}
}

```

Results on CrowdPose test with YOLOv3 human detector

33.18 Halpe (CVPR'2020)

33.18.1 Topdown Heatmap + Hrnet + Dark on Halpe

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and
↪Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},
  year={2020}
}
```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

33.19 OCHuman (CVPR'2019)

33.19.1 Topdown Heatmap + Resnet on Ochuman

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
}
```

(continues on next page)

(continued from previous page)

```

year={2019}
}

```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

33.19.2 Topdown Heatmap + Hrnet on Ochuman

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2019pose2seg,
  title={Pose2seg: Detection free human instance segmentation},
  author={Zhang, Song-Hai and Li, Ruilong and Dong, Xin and Rosin, Paul and Cai, Zixi
↪and Han, Xi and Yang, Dingcheng and Huang, Haozhi and Hu, Shi-Min},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={889--898},
  year={2019}
}

```

Results on OCHuman test dataset with ground-truth bounding boxes

Following the common setting, the models are trained on COCO train dataset, and evaluate on OCHuman dataset.

33.20 AP-10K (NeurIPS'2021)

33.20.1 Topdown Heatmap + Hrnet on Ap10k

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng
↪Tao},

```

(continues on next page)

(continued from previous page)

```

    year={2021},
    eprint={2108.12617},
    archivePrefix={arXiv},
    primaryClass={cs.CV}
}

```

Results on AP-10K validation set

33.20.2 Topdown Heatmap + Resnet on Ap10k

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@misc{yu2021ap10k,
  title={AP-10K: A Benchmark for Animal Pose Estimation in the Wild},
  author={Hang Yu and Yufei Xu and Jing Zhang and Wei Zhao and Ziyu Guan and Dacheng_
↪Tao},
  year={2021},
  eprint={2108.12617},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```

Results on AP-10K validation set

33.21 Desert Locust (Elife'2019)

33.21.1 Topdown Heatmap + Resnet on Locust

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation_
↪using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,_
↪Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},

```

(continues on next page)

(continued from previous page)

```

volume={8},
pages={e47994},
year={2019},
publisher={eLife Sciences Publications Limited}
}

```

Results on Desert Locust test set

33.22 COCO-WholeBody-Face (ECCV'2020)

33.22.1 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_face

```

@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

33.22.2 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_face

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},

```

(continues on next page)

(continued from previous page)

```

pages={770--778},
year={2016}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

33.22.3 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_face

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

33.22.4 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_face

```

@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}

```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

33.22.5 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_face

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

33.22.6 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},
  pages={7093--7102},
}
```

(continues on next page)

(continued from previous page)

```

    year={2020}
}

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Face val set

33.23 MPI-INF-3DHP (3DV'2017)

33.23.1 Pose Lift + Simplebaseline3d on Mpi_inf_3dhp

```

@inproceedings{martinez_2017_3dbaseline,
  title={A simple yet effective baseline for 3d human pose estimation},
  author={Martinez, Julieta and Hossain, Rayat and Romero, Javier and Little, James J.},
  booktitle={ICCV},
  year={2017}
}

@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and ↵
↵Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
↵,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
  year = {2017},
  organization={IEEE},
  doi={10.1109/3dv.2017.000064},
}

```

Results on MPI-INF-3DHP dataset with ground truth 2D detections

1 Differing from the original paper, we didn't apply the max-norm constraint because we found this led to a better convergence and performance.

33.23.2 Video Pose Lift + Videopose3d on Mpi_inf_3dhp

```
@inproceedings{pavullo20193d,
  title={3d human pose estimation in video with temporal convolutions and semi-
  ↪supervised training},
  author={Pavullo, Dario and Feichtenhofer, Christoph and Grangier, David and Auli, ↪
  ↪Michael},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↪
  ↪Recognition},
  pages={7753--7762},
  year={2019}
}
```

```
@inproceedings{mono-3dhp2017,
  author = {Mehta, Dushyant and Rhodin, Helge and Casas, Dan and Fua, Pascal and ↪
  ↪Sotnychenko, Oleksandr and Xu, Weipeng and Theobalt, Christian},
  title = {Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision}
  ↪,
  booktitle = {3D Vision (3DV), 2017 Fifth International Conference on},
  url = {http://gvv.mpi-inf.mpg.de/3dhp_dataset},
  year = {2017},
  organization={IEEE},
  doi={10.1109/3dv.2017.00064},
}
```

Results on MPI-INF-3DHP dataset with ground truth 2D detections, supervised training

33.24 AI Challenger (ArXiv'2017)

33.24.1 Associative Embedding + Hrnet on Aic

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↪
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
```

(continues on next page)

(continued from previous page)

```

    author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
    journal={arXiv preprint arXiv:1711.06475},
    year={2017}
}

```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.24.2 Associative Embedding + Higherhrnet on Aic

```

@inproceedings{newell2017associative,
    title={Associative embedding: End-to-end learning for joint detection and grouping},
    author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
    booktitle={Advances in neural information processing systems},
    pages={2277--2287},
    year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
    title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
    author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
    booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
    pages={5386--5395},
    year={2020}
}

```

```

@article{wu2017ai,
    title={Ai challenger: A large-scale dataset for going deeper in image understanding},
    author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
    journal={arXiv preprint arXiv:1711.06475},
    year={2017}
}

```

Results on AIC validation set without multi-scale test

Results on AIC validation set with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.24.3 Topdown Heatmap + Hrnet on Aic

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC val set with ground-truth bounding boxes

33.24.4 Topdown Heatmap + Resnet on Aic

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{wu2017ai,
  title={Ai challenger: A large-scale dataset for going deeper in image understanding},
  author={Wu, Jiahong and Zheng, He and Zhao, Bo and Li, Yixin and Yan, Baoming and
↪Liang, Rui and Wang, Wenjia and Zhou, Shipei and Lin, Guosen and Fu, Yanwei and others}
↪,
  journal={arXiv preprint arXiv:1711.06475},
  year={2017}
}
```

Results on AIC val set with ground-truth bounding boxes

33.25 MPII-TRB (ICCV'2019)

33.25.1 Topdown Heatmap + Resnet + Mpii on Mpii_trb

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{duan2019trb,
  title={TRB: A Novel Triplet Representation for Understanding 2D Human Body},
  author={Duan, Haodong and Lin, Kwan-Yee and Jin, Sheng and Liu, Wentao and Qian, Chen
↪and Ouyang, Wanli},
  booktitle={Proceedings of the IEEE International Conference on Computer Vision},
  pages={9479--9488},
  year={2019}
}
```

Results on MPII-TRB val set

33.26 WFLW (CVPR'2018)

33.26.1 Deeppose + Resnet on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
```

(continues on next page)

(continued from previous page)

```

    pages={770--778},
    year={2016}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

33.26.2 Deeppose + Resnet + Softwingloss on WFLW

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={770--778},
  year={2016}
}

```

```

@article{lin2021structure,
  title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
  author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and Lu, Jiwen and Zhou, Jie},
  journal={IEEE Transactions on Image Processing},
  year={2021},
  publisher={IEEE}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

33.26.3 Deeppose + Resnet + Wingloss on WFLW

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
pages={1653--1660},
year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
pages={770--778},
year={2016}
}

```

```

@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural_
↪Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↪ Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},
  year={2018},
  pages = {2235-2245},
  organization={IEEE}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,_
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

33.26.4 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},
  author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
  booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
  pages={6971--6981},
  year={2019}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

33.26.5 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
pages={7093--7102},
year={2020}
}

```

```

@inproceedings{wu2018look,
title={Look at boundary: A boundary-aware face alignment algorithm},
author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,↪
↪Qiang},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

33.26.6 Topdown Heatmap + Hrnetv2 on WFLW

```

@article{WangSCJDZLMTWLX19,
title={Deep High-Resolution Representation Learning for Visual Recognition},
author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
journal={TPAMI},
year={2019}
}

```

```

@inproceedings{wu2018look,
title={Look at boundary: A boundary-aware face alignment algorithm},
author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,↪
↪Qiang},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={2129--2138},
year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

33.27 DeepFashion2 (CVPR'2019)

33.27.1 Topdown Heatmap + Resnet on Deepfashion2

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{DeepFashion2,
  author = {Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang
↪and Ping Luo},
  title={A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-
↪Identification of Clothing Images},
  journal={CVPR},
  year={2019}
}
```

Results on DeepFashion2 val set

33.28 PoseTrack18 (CVPR'2018)

33.28.1 Topdown Heatmap + Hrnet on Posetrack18

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andriluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
pages={5167--5176},
year={2018}
}

```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

33.28.2 Topdown Heatmap + Resnet on Posetrack18

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={770--778},
year={2016}
}

```

```

@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andriluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
pages={5167--5176},
year={2018}
}

```

Results on PoseTrack2018 val with ground-truth bounding boxes

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

Results on PoseTrack2018 val with [MMDetection](#) pre-trained [Cascade R-CNN](#) (X-101-64x4d-FPN) human detector

The models are first pre-trained on COCO dataset, and then fine-tuned on PoseTrack18.

33.28.3 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,
  title = {Learning Temporal Pose Estimation from Sparsely Labeled Videos},
  author = {Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo,
    ↪and Torresani, Lorenzo},
  booktitle = {Advances in Neural Information Processing Systems 33},
  year = {2019},
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
    ↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
    ↪Recognition},
  pages={5167--5176},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
    ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)¹.

¹ Please download the precomputed human bounding boxes on PoseTrack2018 val from \$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json and place it here: \$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json to be consistent with the [config](#). Please refer to DATA

Preparation for more detail about data preparation.

33.29 COCO-WholeBody-Hand (ECCV'2020)

33.29.1 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪ Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.29.2 Topdown Heatmap + Mobilenetv2 + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪ Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.29.3 Topdown Heatmap + Litehrnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and ↵
↵Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.29.4 Topdown Heatmap + Hourglass + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian, ↵
↵Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```


Results on COCO-WholeBody-Hand val set

33.29.5 Topdown Heatmap + Scnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
  ↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.29.6 Topdown Heatmap + Resnet + Coco + Wholebody on Coco_wholebody_hand

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

(continues on next page)

(continued from previous page)

```
year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.29.7 Topdown Heatmap + Hrnetv2 + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
    Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Hand val set

33.30 COFW (ICCV'2013)

33.30.1 Topdown Heatmap + Hrnetv2 on Cofw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{burgos2013robust,
  title={Robust face landmark estimation under occlusion},
  author={Burgos-Artizzu, Xavier P and Perona, Pietro and Doll{\`a}r, Piotr},
  booktitle={Proceedings of the IEEE international conference on computer vision},
  pages={1513--1520},
  year={2013}
}
```

Results on COFW dataset

The model is trained on COFW train.

33.31 DeepFashion (CVPR'2016)

33.31.1 Deeppose + Resnet on Deepfashion

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuQWTCvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↪Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↪(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWTEccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

33.31.2 Topdown Heatmap + Resnet on Deepfashion

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{liuQWtcvpr16DeepFashion,
  author = {Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou},
  title = {DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich
↵Annotations},
  booktitle = {Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
↵(CVPR)},
  month = {June},
  year = {2016}
}
```

```
@inproceedings{liuYLWteccv16FashionLandmark,
  author = {Liu, Ziwei and Yan, Sijie and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou},
  title = {Fashion Landmark Detection in the Wild},
  booktitle = {European Conference on Computer Vision (ECCV)},
  month = {October},
  year = {2016}
}
```

Results on DeepFashion val set

33.32 COCO-WholeBody (ECCV'2020)

33.32.1 Associative Embedding + Hrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

33.32.2 Associative Embedding + Higherhrnet on Coco-Wholebody

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose
↪Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,
↪Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val without multi-scale test

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

33.32.3 Topdown Heatmap + Resnet on Coco-Wholebody

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

33.32.4 Topdown Heatmap + Hrnet on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

33.32.5 Topdown Heatmap + Tcformer on Coco-Wholebody

```
@inproceedings{zeng2022not,
  title={Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering},
  ↪Transformer},
  author={Zeng, Wang and Jin, Sheng and Liu, Wentao and Qian, Chen and Luo, Ping and
  ↪Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  ↪Recognition},
  pages={11101--11111},
  year={2022}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

33.32.6 Topdown Heatmap + Vipnas on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
  ↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  year={2021}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

33.32.7 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

33.32.8 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```



```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
  ↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

33.33 Horse-10 (WACV'2021)

33.33.1 Topdown Heatmap + Resnet on Horse10

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekgonul,
  ↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
  ↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

33.33.2 Topdown Heatmap + Hrnet on Horse10

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{mathis2021pretraining,
  title={Pretraining boosts out-of-domain robustness for pose estimation},
  author={Mathis, Alexander and Biasi, Thomas and Schneider, Steffen and Yuksekogul,
↪Mert and Rogers, Byron and Bethge, Matthias and Mathis, Mackenzie W},
  booktitle={Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
↪Vision},
  pages={1859--1868},
  year={2021}
}
```

Results on Horse-10 test set

33.34 Grévy's Zebra (Elife'2019)

33.34.1 Topdown Heatmap + Resnet on Zebra

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@article{graving2019deepposekit,
  title={DeepPoseKit, a software toolkit for fast and robust animal pose estimation
↪using deep learning},
  author={Graving, Jacob M and Chae, Daniel and Naik, Hemal and Li, Liang and Koger,
↪Benjamin and Costelloe, Blair R and Couzin, Iain D},
  journal={Elife},
  volume={8},
  pages={e47994},
  year={2019},
  publisher={eLife Sciences Publications Limited}
}
```

Results on Grévy's Zebra test set

33.35 COCO (ECCV'2014)

33.35.1 Associative Embedding + Higherhrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
```

(continues on next page)

(continued from previous page)

```

    year={2017}
}

```

```

@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↪ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↪ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↪ Recognition},
  pages={5386--5395},
  year={2020}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

33.35.2 Associative Embedding + Mobilenetv2 on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and_
↪ Chen, Liang-Chieh},

```

(continues on next page)

(continued from previous page)

```

booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={4510--4520},
year={2018}
}

```

```

@inproceedings{lin2014microsoft,
title={Microsoft coco: Common objects in context},
author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.35.3 Associative Embedding + Hrnet on Coco

```

@inproceedings{newell2017associative,
title={Associative embedding: End-to-end learning for joint detection and grouping},
author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
booktitle={Advances in neural information processing systems},
pages={2277--2287},
year={2017}
}

```

```

@inproceedings{sun2019deep,
title={Deep high-resolution representation learning for human pose estimation},
author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
pages={5693--5703},
year={2019}
}

```

```

@inproceedings{lin2014microsoft,
title={Microsoft coco: Common objects in context},
author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.35.4 Associative Embedding + Higherhrnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose ↵
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang, ↵
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern ↵
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and ↵
↵ Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.35.5 Associative Embedding + Resnet on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern ↵
↵ recognition},
  pages={770--778},
}
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.35.6 Associative Embedding + Hrnet + Udp on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},

```

(continues on next page)

(continued from previous page)

```

booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

33.35.7 Associative Embedding + Hourglass + Ae on Coco

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

33.35.8 Deeppose + Resnet on Coco

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szedgy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.9 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```



```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.10 Dekr + Hrnet on Coco

```
@inproceedings{geng2021bottom,
  title={Bottom-up human pose estimation via disentangled keypoint regression},
  author={Geng, Zigang and Sun, Ke and Xiao, Bin and Zhang, Zhaoxiang and Wang, Jingdong}
↪ ,
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={14676--14686},
  year={2021}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

Results on COCO val2017 with multi-scale test. 3 default scales ([2, 1, 0.5]) are used

* these configs are generally used for evaluation. The training settings are identical to their single-scale counterparts.

The results of models provided by the authors on COCO val2017 using the same evaluation protocol

The discrepancy between these results and that shown in paper is attributed to the differences in implementation details in evaluation process.

33.35.11 Topdown Heatmap + Seresnet on Coco

```
@inproceedings{hu2018squeeze,
  title={Squeeze-and-excitation networks},
  author={Hu, Jie and Shen, Li and Sun, Gang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={7132--7141},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that * means without imagenet pre-training.

33.35.12 Topdown Heatmap + Resnetv1d on Coco

```
@inproceedings{he2019bag,
  title={Bag of tricks for image classification with convolutional neural networks},
  author={He, Tong and Zhang, Zhi and Zhang, Hang and Zhang, Zhongyue and Xie, Junyuan
↪and Li, Mu},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={558--567},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.13 Topdown Heatmap + Hourglass on Coco

```
@inproceedings{newell2016stacked,
  title={Stacked hourglass networks for human pose estimation},
  author={Newell, Alejandro and Yang, Kaiyu and Deng, Jia},
  booktitle={European conference on computer vision},
  pages={483--499},
  year={2016},
  organization={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.14 Topdown Heatmap + RSN on Coco

```
@misc{cai2020learning,
  title={Learning Delicate Local Representations for Multi-Person Pose Estimation},
  author={Yuanhao Cai and Zhicheng Wang and Zhengxiong Luo and Binyi Yin and Angang Du
  ↪and Haoqian Wang and Xinyu Zhou and Erjin Zhou and Xiangyu Zhang and Jian Sun},
  year={2020},
  eprint={2003.04030},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.15 Topdown Heatmap + Resnet + Fp16 on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↵and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↵Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.16 Topdown Heatmap + Mobilenetv2 on Coco

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↵Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.17 Topdown Heatmap + Shufflenetv1 on Coco

```
@inproceedings{zhang2018shufflenet,
  title={Shufflenet: An extremely efficient convolutional neural network for mobile
↪devices},
  author={Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={6848--6856},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.18 Topdown Heatmap + MSPN on Coco

```
@article{li2019rethinking,
  title={Rethinking on Multi-Stage Networks for Human Pose Estimation},
  author={Li, Wenbo and Wang, Zhicheng and Yin, Binyi and Peng, Qixiang and Du, Yuming
↪and Xiao, Tianzi and Yu, Gang and Lu, Hongtao and Wei, Yichen and Sun, Jian},
  journal={arXiv preprint arXiv:1901.00148},
  year={2019}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},

```

(continues on next page)

(continued from previous page)

```
booktitle={European conference on computer vision},
pages={740--755},
year={2014},
organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.19 Topdown Heatmap + Hrnet + Fp16 on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory_
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and_
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.20 Topdown Heatmap + Hrnet on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↪recognition},
  pages={5693--5703},
```

(continues on next page)

(continued from previous page)

```

year={2019}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.21 Topdown Heatmap + Resnext on Coco

```

@inproceedings{xie2017aggregated,
  title={Aggregated residual transformations for deep neural networks},
  author={Xie, Saining and Girshick, Ross and Doll{'a}r, Piotr and Tu, Zhuowen and He,
↪Kaiming},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1492--1500},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.22 Topdown Heatmap + Resnet + Dark on Coco

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.23 Topdown Heatmap + VGG on Coco

```
@article{simonyan2014very,
  title={Very deep convolutional networks for large-scale image recognition},
  author={Simonyan, Karen and Zisserman, Andrew},
  journal={arXiv preprint arXiv:1409.1556},
  year={2014}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.24 Topdown Heatmap + Shufflenetv2 on Coco

```
@inproceedings{ma2018shufflenet,
  title={Shufflenet v2: Practical guidelines for efficient cnn architecture design},
  author={Ma, Ningning and Zhang, Xiangyu and Zheng, Hai-Tao and Sun, Jian},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={116--131},
  year={2018}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.25 Topdown Heatmap + Hrnet + Augmentation on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
  ↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{buslaev2020albumentations,
  title={Albumentations: fast and flexible image augmentations},
  author={Buslaev, Alexander and Iglovikov, Vladimir I and Khvedchenya, Eugene and
  ↪Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A},
  journal={Information},
  volume={11},
  number={2},
  pages={125},
  year={2020},
  publisher={Multidisciplinary Digital Publishing Institute}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
  ↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
}
```

(continues on next page)

(continued from previous page)

```

year={2014},
organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.26 Topdown Heatmap + Swin on Coco

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{liu2021swin,
  title={Swin transformer: Hierarchical vision transformer using shifted windows},
  author={Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang,
↪Zheng and Lin, Stephen and Guo, Baining},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={10012--10022},
  year={2021}
}

```

```

@inproceedings{lin2017feature,
  title={Feature pyramid networks for object detection},
  author={Lin, Tsung-Yi and Doll{\`a}r, Piotr and Girshick, Ross and He, Kaiming and
↪Hariharan, Bharath and Belongie, Serge},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2117--2125},
  year={2017}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.27 Topdown Heatmap + Litehrnet on Coco

```
@inproceedings{Yulitehrnet21,
  title={Lite-HRNet: A Lightweight High-Resolution Network},
  author={Yu, Changqian and Xiao, Bin and Gao, Changxin and Yuan, Lu and Zhang, Lei and
↪Sang, Nong and Wang, Jingdong},
  booktitle={CVPR},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.28 Topdown Heatmap + Snet on Coco

```
@inproceedings{liu2020improving,
  title={Improving Convolutional Networks with Self-Calibrated Convolutions},
  author={Liu, Jiang-Jiang and Hou, Qibin and Cheng, Ming-Ming and Wang, Changhu and
↪Feng, Jiashi},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={10096--10105},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.29 Topdown Heatmap + Hrformer on Coco

```
@article{yuan2021hrformer,
  title={HRFormer: High-Resolution Vision Transformer for Dense Predict},
  author={Yuan, Yuhui and Fu, Rao and Huang, Lang and Lin, Weihong and Zhang, Chao and
↪Chen, Xilin and Wang, Jingdong},
  journal={Advances in Neural Information Processing Systems},
  volume={34},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.30 Topdown Heatmap + Resnest on Coco

```
@article{zhang2020resnest,
  title={ResNeSt: Split-Attention Networks},
  author={Zhang, Hang and Wu, Chongruo and Zhang, Zhongyue and Zhu, Yi and Zhang, Zhi
↪and Lin, Haibin and Sun, Yue and He, Tong and Muller, Jonas and Manmatha, R. and Li,
↪Mu and Smola, Alexander},
  journal={arXiv preprint arXiv:2004.08955},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.31 Topdown Heatmap + PVT on Coco

```
@inproceedings{wang2021pyramid,
  title={Pyramid vision transformer: A versatile backbone for dense prediction without
↪convolutions},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao
↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={568--578},
  year={2021}
}
```

```
@article{wang2022pvt,
  title={PVT v2: Improved baselines with Pyramid Vision Transformer},
  author={Wang, Wenhai and Xie, Enze and Li, Xiang and Fan, Deng-Ping and Song, Kaitao
↪and Liang, Ding and Lu, Tong and Luo, Ping and Shao, Ling},
  journal={Computational Visual Media},
  pages={1--10},
  year={2022},
  publisher={Springer}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.32 Topdown Heatmap + Resnet on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
}
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.33 Topdown Heatmap + Alexnet on Coco

```

@inproceedings{krizhevsky2012imagenet,
  title={Imagenet classification with deep convolutional neural networks},
  author={Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E},
  booktitle={Advances in neural information processing systems},
  pages={1097--1105},
  year={2012}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.34 Topdown Heatmap + Hrnet + Udp on Coco

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of [DARK](#).

33.35.35 Topdown Heatmap + Hrnet + Dark on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
}
```

(continues on next page)

(continued from previous page)

```
organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.36 Topdown Heatmap + Vipnas on Coco

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  year={2021}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.37 Topdown Heatmap + CPM on Coco

```
@inproceedings{wei2016convolutional,
  title={Convolutional pose machines},
  author={Wei, Shih-En and Ramakrishna, Varun and Kanade, Takeo and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↪ Recognition},
  pages={4724--4732},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

33.35.38 Cid + Hrnet on Coco

```
@InProceedings{Wang_2022_CVPR,
  author    = {Wang, Dongkai and Zhang, Shiliang},
  title     = {Contextual Instance Decoupling for Robust Multi-Person Pose Estimation},
  booktitle = {Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition (CVPR)},
  month     = {June},
  year      = {2022},
  pages     = {11060-11068}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

33.35.39 Posewarper + Hrnet + Posetrack18 on Posetrack18

```
@inproceedings{NIPS2019_gberta,
  title = {Learning Temporal Pose Estimation from Sparsely Labeled Videos},
  author = {Bertasius, Gedas and Feichtenhofer, Christoph, and Tran, Du and Shi, Jianbo,
↪and Torresani, Lorenzo},
  booktitle = {Advances in Neural Information Processing Systems 33},
  year = {2019},
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{andrilluka2018posetrack,
  title={Posetrack: A benchmark for human pose estimation and tracking},
  author={Andrilluka, Mykhaylo and Iqbal, Umar and Insafutdinov, Eldar and Pishchulin,
↪Leonid and Milan, Anton and Gall, Juergen and Schiele, Bernt},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern
↪Recognition},
  pages={5167--5176},
}
```

(continues on next page)

(continued from previous page)

```

    year={2018}
}

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Note that the training of PoseWarper can be split into two stages.

The first-stage is trained with the pre-trained [checkpoint](#) on COCO dataset, and the main backbone is fine-tuned on PoseTrack18 in a single-frame setting.

The second-stage is trained with the last [checkpoint](#) from the first stage, and the warping offsets are learned in a multi-frame setting while the backbone is frozen.

Results on PoseTrack2018 val with ground-truth bounding boxes

Results on PoseTrack2018 val with precomputed human bounding boxes from PoseWarper supplementary data files from [this link](#)¹.

¹ Please download the precomputed human bounding boxes on PoseTrack2018 val from \$PoseWarper_supp_files/posetrack18_precomputed_boxes/val_boxes.json and place it here: \$mmpose/data/posetrack18/posetrack18_precomputed_boxes/val_boxes.json to be consistent with the [config](#). Please refer to DATA Preparation for more detail about data preparation.

33.36 RHD (ICCV'2017)

33.36.1 Deeppose + Resnet on Rhd2d

```

@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}

```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

33.36.2 Topdown Heatmap + Mobilenetv2 on Rhd2d

```
@inproceedings{sandler2018mobilenetv2,
  title={Mobilenetv2: Inverted residuals and linear bottlenecks},
  author={Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and
↪Chen, Liang-Chieh},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={4510--4520},
  year={2018}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

33.36.3 Topdown Heatmap + Resnet on Rhd2d

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
}
```

(continues on next page)

(continued from previous page)

```

    year={2016}
}

```

```

@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

33.36.4 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}

```

```

@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}

```

Results on RHD test set

33.36.5 Topdown Heatmap + Hrnetv2 + Dark on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
  Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

33.36.6 Topdown Heatmap + Hrnetv2 on Rhd2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

34.1 RLE (ICCV'2021)

34.1.1 Deeppose + Resnet + Rle on Coco

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
}
```

(continues on next page)

(continued from previous page)

```
organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.1.2 Deeppose + Resnet + Rle on Mpii

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{li2021human,
  title={Human pose regression with residual log-likelihood estimation},
  author={Li, Jiefeng and Bian, Siyuan and Zeng, Ailing and Wang, Can and Pang, Bo and
↪Liu, Wentao and Lu, Cewu},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={11025--11034},
  year={2021}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

34.2 SoftWingloss (TIP'2021)

34.2.1 Deeppose + Resnet + Softwingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{lin2021structure,
  title={Structure-Coherent Deep Feature Learning for Robust Face Alignment},
  author={Lin, Chunze and Zhu, Beier and Wang, Quan and Liao, Renjie and Qian, Chen and
↪Lu, Jiwen and Zhou, Jie},
  journal={IEEE Transactions on Image Processing},
  year={2021},
  publisher={IEEE}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

34.3 FP16 (ArXiv'2017)

34.3.1 Topdown Heatmap + Resnet + Fp16 on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={770--778},
  year={2016}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.3.2 Topdown Heatmap + Hrnet + Fp16 on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{micikevicius2017mixed,
  title={Mixed precision training},
  author={Micikevicius, Paulius and Narang, Sharan and Alben, Jonah and Diamos, Gregory
↪and Elsen, Erich and Garcia, David and Ginsburg, Boris and Houston, Michael and
↪Kuchaiev, Oleksii and Venkatesh, Ganesh and others},
  journal={arXiv preprint arXiv:1710.03740},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a'}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.4 Albumentations (Information'2020)

34.4.1 Topdown Heatmap + Hrnet + Augmentation on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@article{buslaev2020albumentations,
  title={Albumentations: fast and flexible image augmentations},
  author={Buslaev, Alexander and Iglovikov, Vladimir I and Khvedchenya, Eugene and
↪Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A},
  journal={Information},

```

(continues on next page)

(continued from previous page)

```

volume={11},
number={2},
pages={125},
year={2020},
publisher={Multidisciplinary Digital Publishing Institute}
}

```

```

@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}

```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.5 AdaptiveWingloss (ICCV'2019)

34.5.1 Topdown Heatmap + Hrnetv2 + Awing on WFLW

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{wang2019adaptive,
  title={Adaptive wing loss for robust face alignment via heatmap regression},
  author={Wang, Xinyao and Bo, Liefeng and Fuxin, Li},
  booktitle={Proceedings of the IEEE/CVF international conference on computer vision},
  pages={6971--6981},
  year={2019}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
↪ Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

34.6 FPN (CVPR'2017)

34.6.1 Topdown Heatmap + Swin on Coco

```
@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}
```

```
@inproceedings{liu2021swin,
  title={Swin transformer: Hierarchical vision transformer using shifted windows},
  author={Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang,
↪Zheng and Lin, Stephen and Guo, Baining},
  booktitle={Proceedings of the IEEE/CVF International Conference on Computer Vision},
  pages={10012--10022},
  year={2021}
}
```

```
@inproceedings{lin2017feature,
  title={Feature pyramid networks for object detection},
  author={Lin, Tsung-Yi and Doll{\`a}r, Piotr and Girshick, Ross and He, Kaiming and
↪Hariharan, Bharath and Belongie, Serge},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={2117--2125},
  year={2017}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{\`a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.7 UDP (CVPR'2020)

34.7.1 Associative Embedding + Higherhrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{cheng2020higherhrnet,
  title={HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose_
↵ Estimation},
  author={Cheng, Bowen and Xiao, Bin and Wang, Jingdong and Shi, Honghui and Huang,_
↵ Thomas S and Zhang, Lei},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern_
↵ Recognition},
  pages={5386--5395},
  year={2020}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵ ,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↵ Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

34.7.2 Associative Embedding + Hrnet + Udp on Coco

```
@inproceedings{newell2017associative,
  title={Associative embedding: End-to-end learning for joint detection and grouping},
  author={Newell, Alejandro and Huang, Zhiao and Deng, Jia},
  booktitle={Advances in neural information processing systems},
  pages={2277--2287},
  year={2017}
}
```

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 without multi-scale test

34.7.3 Topdown Heatmap + Hrnet + Udp on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↵recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and_
↵Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

Note that, UDP also adopts the unbiased encoding/decoding algorithm of [DARK](#).

34.7.4 Topdown Heatmap + Hrnetv2 + Udp on Onehand10k

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
```

(continues on next page)

(continued from previous page)

```

pages={3258--3268},
year={2018},
publisher={IEEE}
}

```

Results on OneHand10K val set

34.7.5 Topdown Heatmap + Hrnetv2 + Udp on Panoptic2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↪ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↪ ,
  month = {June},
  year = {2020}
}

```

```

@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern_
↪ Recognition},
  pages={1145--1153},
  year={2017}
}

```

Results on CMU Panoptic (MPII+NZSL val set)

34.7.6 Topdown Heatmap + Hrnetv2 + Udp on Rhd2d

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```
@InProceedings{Huang_2020_CVPR,
  author = {Huang, Junjie and Zhu, Zheng and Guo, Feng and Huang, Guan},
  title = {The Devil Is in the Details: Delving Into Unbiased Data Processing for Human_
↵ Pose Estimation},
  booktitle = {The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)}
↵,
  month = {June},
  year = {2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

34.8 Wingloss (CVPR'2018)

34.8.1 Deeppose + Resnet + Wingloss on WFLW

```
@inproceedings{toshev2014deeppose,
  title={Deeppose: Human pose estimation via deep neural networks},
  author={Toshev, Alexander and Szegedy, Christian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↵ recognition},
  pages={1653--1660},
  year={2014}
}
```

```
@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern_
↵ recognition},
  pages={770--778},
  year={2016}
}
```

```
@inproceedings{feng2018wing,
  title={Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural_
↵ Networks},
  author={Feng, Zhen-Hua and Kittler, Josef and Awais, Muhammad and Huber, Patrik and Wu,
↵ Xiao-Jun},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on},

```

(continues on next page)

(continued from previous page)

```

year={2018},
pages={2235-2245},
organization={IEEE}
}

```

```

@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou, Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={2129--2138},
  year={2018}
}

```

Results on WFLW dataset

The model is trained on WFLW train.

34.9 DarkPose (CVPR'2020)

34.9.1 Topdown Heatmap + Resnet + Dark on Coco

```

@inproceedings{xiao2018simple,
  title={Simple baselines for human pose estimation and tracking},
  author={Xiao, Bin and Wu, Haiping and Wei, Yichen},
  booktitle={Proceedings of the European conference on computer vision (ECCV)},
  pages={466--481},
  year={2018}
}

```

```

@inproceedings{he2016deep,
  title={Deep residual learning for image recognition},
  author={He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern recognition},
  pages={770--778},
  year={2016}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},
  pages={7093--7102},
  year={2020}
}

```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.9.2 Topdown Heatmap + Hrnet + Dark on Coco

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{lin2014microsoft,
  title={Microsoft coco: Common objects in context},
  author={Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and
↪Perona, Pietro and Ramanan, Deva and Doll{'a}r, Piotr and Zitnick, C Lawrence},
  booktitle={European conference on computer vision},
  pages={740--755},
  year={2014},
  organization={Springer}
}
```

Results on COCO val2017 with detector having human AP of 56.4 on COCO val2017 dataset

34.9.3 Topdown Heatmap + Hrnet + Dark on Mpii

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{andrilluka14cvpr,
  author = {Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt}
↪,
  title = {2D Human Pose Estimation: New Benchmark and State of the Art Analysis},
  booktitle = {IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  year = {2014},
  month = {June}
}
```

Results on MPII val set

34.9.4 Topdown Heatmap + Hrnetv2 + Dark on Aflw

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
        Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
        Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{koestinger2011annotated,
  title={Annotated facial landmarks in the wild: A large-scale, real-world database for
↪facial landmark localization},
  author={Koestinger, Martin and Wohlhart, Paul and Roth, Peter M and Bischof, Horst},
  booktitle={2011 IEEE international conference on computer vision workshops (ICCV
↪workshops)},
  pages={2144--2151},
  year={2011},
  organization={IEEE}
}
```

Results on AFLW dataset

The model is trained on AFLW train and evaluated on AFLW full and frontal.

34.9.5 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_face

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody-Face val set

34.9.6 Topdown Heatmap + Hrnetv2 + Dark on WFLW

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{wu2018look,
  title={Look at boundary: A boundary-aware face alignment algorithm},
  author={Wu, Wayne and Qian, Chen and Yang, Shuo and Wang, Quan and Cai, Yici and Zhou,
    Qiang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
    recognition},
  pages={2129--2138},
  year={2018}
}
```

Results on WFLW dataset

The model is trained on WFLW train.

34.9.7 Topdown Heatmap + Hrnetv2 + Dark + Coco + Wholebody on Coco_wholebody_hand

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
    Recognition},

```

(continues on next page)

(continued from previous page)

```

    pages={7093--7102},
    year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody-Hand val set

34.9.8 Topdown Heatmap + Hrnetv2 + Dark on Onehand10k

```

@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@article{wang2018mask,
  title={Mask-pose cascaded cnn for 2d hand pose estimation from single color image},
  author={Wang, Yangang and Peng, Cong and Liu, Yebin},
  journal={IEEE Transactions on Circuits and Systems for Video Technology},
  volume={29},
  number={11},
  pages={3258--3268},
  year={2018},
  publisher={IEEE}
}

```

Results on OneHand10K val set

34.9.9 Topdown Heatmap + Hrnetv2 + Dark on Panoptic2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{simon2017hand,
  title={Hand keypoint detection in single images using multiview bootstrapping},
  author={Simon, Tomas and Joo, Hanbyul and Matthews, Iain and Sheikh, Yaser},
  booktitle={Proceedings of the IEEE conference on Computer Vision and Pattern
↵Recognition},
  pages={1145--1153},
  year={2017}
}
```

Results on CMU Panoptic (MPII+NZSL val set)

34.9.10 Topdown Heatmap + Hrnetv2 + Dark on Rh2d

```
@article{WangSCJDZLMTWLX19,
  title={Deep High-Resolution Representation Learning for Visual Recognition},
  author={Jingdong Wang and Ke Sun and Tianheng Cheng and
    Borui Jiang and Chaorui Deng and Yang Zhao and Dong Liu and Yadong Mu and
    Mingkui Tan and Xinggang Wang and Wenyu Liu and Bin Xiao},
  journal={TPAMI},
  year={2019}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↵Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@TechReport{zb2017hand,
  author={Christian Zimmermann and Thomas Brox},
  title={Learning to Estimate 3D Hand Pose from Single RGB Images},
  institution={arXiv:1705.01389},
  year={2017},
  note="https://arxiv.org/abs/1705.01389",
  url="https://lmb.informatik.uni-freiburg.de/projects/hand3d/"
}
```

Results on RHD test set

34.9.11 Topdown Heatmap + Vipnas + Dark on Coco-Wholebody

```
@article{xu2021vipnas,
  title={ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search},
  author={Xu, Lumin and Guan, Yingda and Jin, Sheng and Liu, Wentao and Qian, Chen and
↪ Luo, Ping and Ouyang, Wanli and Wang, Xiaogang},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  year={2021}
}
```

```
@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪ Recognition},
  pages={7093--7102},
  year={2020}
}
```

```
@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪ Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}
```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

34.9.12 Topdown Heatmap + Hrnet + Dark on Coco-Wholebody

```
@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪ recognition},
  pages={5693--5703},

```

(continues on next page)

(continued from previous page)

```

    year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{jin2020whole,
  title={Whole-Body Human Pose Estimation in the Wild},
  author={Jin, Sheng and Xu, Lumin and Xu, Jin and Wang, Can and Liu, Wentao and Qian,
↪Chen and Ouyang, Wanli and Luo, Ping},
  booktitle={Proceedings of the European Conference on Computer Vision (ECCV)},
  year={2020}
}

```

Results on COCO-WholeBody v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on COCO-WholeBody dataset. We find this will lead to better performance.

34.9.13 Topdown Heatmap + Hrnet + Dark on Halpe

```

@inproceedings{sun2019deep,
  title={Deep high-resolution representation learning for human pose estimation},
  author={Sun, Ke and Xiao, Bin and Liu, Dong and Wang, Jingdong},
  booktitle={Proceedings of the IEEE conference on computer vision and pattern
↪recognition},
  pages={5693--5703},
  year={2019}
}

```

```

@inproceedings{zhang2020distribution,
  title={Distribution-aware coordinate representation for human pose estimation},
  author={Zhang, Feng and Zhu, Xiatian and Dai, Hanbin and Ye, Mao and Zhu, Ce},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
↪Recognition},
  pages={7093--7102},
  year={2020}
}

```

```

@inproceedings{li2020pastanet,
  title={PaStaNet: Toward Human Activity Knowledge Engine},
  author={Li, Yong-Lu and Xu, Liang and Liu, Xinpeng and Huang, Xijie and Xu, Yue and
↪Wang, Shiyi and Fang, Hao-Shu and Ma, Ze and Chen, Mingyang and Lu, Cewu},
  booktitle={CVPR},

```

(continues on next page)

(continued from previous page)

```
year={2020}  
}
```

Results on Halpe v1.0 val with detector having human AP of 56.4 on COCO val2017 dataset

Note: + means the model is first pre-trained on original COCO dataset, and then fine-tuned on Halpe dataset. We find this will lead to better performance.

TUTORIAL 0: LEARN ABOUT CONFIGS

We use python files as configs, incorporate modular and inheritance design into our config system, which is convenient to conduct various experiments. You can find all the provided configs under `$MMPose/configs`. If you wish to inspect the config file, you may run `python tools/analysis/print_config.py /PATH/TO/CONFIG` to see the complete config.

- *Modify config through script arguments*
- *Config File Naming Convention*
 - Config System Example
- *FAQ*
 - *Use intermediate variables in configs*

35.1 Modify config through script arguments

When submitting jobs using “tools/train.py” or “tools/test.py”, you may specify `--cfg-options` to in-place modify the config.

- Update config keys of dict chains.

The config options can be specified following the order of the dict keys in the original config. For example, `--cfg-options model.backbone.norm_eval=False` changes the all BN modules in model backbones to train mode.

- Update keys inside a list of configs.

Some config dicts are composed as a list in your config. For example, the training pipeline `data.train.pipeline` is normally a list e.g. `[dict(type='LoadImageFromFile'), dict(type='TopDownRandomFlip', flip_prob=0.5), ...]`. If you want to change 'flip_prob=0.5' to 'flip_prob=0.0' in the pipeline, you may specify `--cfg-options data.train.pipeline.1.flip_prob=0.0`.

- Update values of list/tuples.

If the value to be updated is a list or a tuple. For example, the config file normally sets `workflow=[('train', 1)]`. If you want to change this key, you may specify `--cfg-options workflow="[(train,1),(val,1)]"`. Note that the quotation mark " is necessary to support list/tuple data types, and that **NO** white space is allowed inside the quotation marks in the specified value.

35.2 Config File Naming Convention

We follow the style below to name config files. Contributors are advised to follow the same style.

```
configs/{topic}/{task}/{algorithm}/{dataset}/{backbone}_{model_setting}_{dataset}_{input_
↪size}_{technique}.py
```

{xxx} is required field and [yyy] is optional.

- {topic}: topic type, e.g. body, face, hand, animal, etc.
- {task}: task type, [2d | 3d]_[kpt | mesh]_[sview | mview]_[rgb | rgbd]_[img | vid]. The task is categorized in 5: (1) 2D or 3D pose estimation, (2) representation type: keypoint (kpt), mesh, or DensePose (dense). (3) Single-view (sview) or multi-view (mview), (4) RGB or RGBD, and (5) Image (img) or Video (vid). e.g. 2d_kpt_sview_rgb_img, 3d_kpt_sview_rgb_vid, etc.
- {algorithm}: algorithm type, e.g. associative_embedding, deeppose, etc.
- {dataset}: dataset name, e.g. coco, etc.
- {backbone}: backbone type, e.g. res50 (ResNet-50), etc.
- [model_setting]: specific setting for some models.
- [input_size]: input size of the model.
- [technique]: some specific techniques, including losses, augmentation and tricks, e.g. wingloss, udp, fp16.

35.2.1 Config System

- An Example of 2D Top-down Heatmap-based Human Pose Estimation

To help the users have a basic idea of a complete config structure and the modules in the config system, we make brief comments on ‘https://github.com/open-mmlab/mmpose/tree/e1ec589884235bee875c89102170439a991f8450/configs/top_down/resnet/coco/res50_coco_256x192.py’ as the following. For more detailed usage and alternative for per parameter in each module, please refer to the API documentation.

```
# runtime settings
log_level = 'INFO' # The level of logging
load_from = None # load models as a pre-trained model from a given path. This will
↪not resume training
resume_from = None # Resume checkpoints from a given path, the training will be
↪resumed from the epoch when the checkpoint's is saved
dist_params = dict(backend='nccl') # Parameters to setup distributed training, the
↪port can also be set
workflow = [('train', 1)] # Workflow for runner. [('train', 1)] means there is only
↪one workflow and the workflow named 'train' is executed once
checkpoint_config = dict( # Config to set the checkpoint hook, Refer to https://
↪github.com/open-mmlab/mmcv/blob/master/mmcv/runner/hooks/checkpoint.py for
↪implementation
    interval=10) # Interval to save checkpoint
evaluation = dict( # Config of evaluation during training
    interval=10, # Interval to perform evaluation
    metric='mAP', # Metrics to be performed
    save_best='AP') # set `AP` as key indicator to save best checkpoint
```

(continues on next page)

(continued from previous page)

```

# optimizer
optimizer = dict(
    # Config used to build optimizer, support (1). All the optimizers in PyTorch
    # whose arguments are also the same as those in PyTorch. (2). Custom optimizers
    # which are builed on `constructor`, referring to "tutorials/4_new_modules.md"
    # for implementation.
    type='Adam', # Type of optimizer, refer to https://github.com/open-mmlab/mmcv/
    ↪blob/master/mmcv/runner/optimizer/default_constructor.py#L13 for more details
    lr=5e-4, # Learning rate, see detail usages of the parameters in the
    ↪documentation of PyTorch
)
optimizer_config = dict(grad_clip=None) # Do not use gradient clip
# learning policy
lr_config = dict( # Learning rate scheduler config used to register LrUpdater hook
    policy='step', # Policy of scheduler, also support CosineAnnealing, Cyclic,
    ↪etc. Refer to details of supported LrUpdater from https://github.com/open-mmlab/
    ↪mmcv/blob/master/mmcv/runner/hooks/lr_updater.py#L9
    warmup='linear', # Type of warmup used. It can be None(use no warmup), 'constant',
    ↪'linear' or 'exp'.
    warmup_iters=500, # The number of iterations or epochs that warmup
    warmup_ratio=0.001, # LR used at the beginning of warmup equals to warmup
    ↪ratio * initial_lr
    step=[170, 200]) # Steps to decay the learning rate
total_epochs = 210 # Total epochs to train the model
log_config = dict( # Config to register logger hook
    interval=50, # Interval to print the log
    hooks=[
        dict(type='TextLoggerHook'), # The logger used to record the training
    ↪process
        # dict(type='TensorboardLoggerHook') # The Tensorboard logger is also
    ↪supported
    ])

channel_cfg = dict(
    num_output_channels=17, # The output channels of keypoint head
    dataset_joints=17, # Number of joints in the dataset
    dataset_channel=[ # Dataset supported channels
        [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16],
    ],
    inference_channel=[ # Channels to output
        0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
    ])

# model settings
model = dict( # Config of the model
    type='TopDown', # Type of the model
    pretrained='torchvision://resnet50', # The url/site of the pretrained model
    backbone=dict( # Dict for backbone
        type='ResNet', # Name of the backbone
        depth=50), # Depth of ResNet model
    keypoint_head=dict( # Dict for keypoint head
        type='TopdownHeatmapSimpleHead', # Name of keypoint head

```

(continues on next page)

(continued from previous page)

```

        in_channels=2048, # The input channels of keypoint head
        out_channels=channel_cfg['num_output_channels'], # The output channels of
↳keypoint head
        loss_keypoint=dict( # Dict for keypoint loss
            type='JointsMSELoss', # Name of keypoint loss
            use_target_weight=True)), # Whether to consider target_weight during
↳loss calculation
        train_cfg=dict(), # Config of training hyper-parameters
        test_cfg=dict( # Config of testing hyper-parameters
            flip_test=True, # Whether to use flip-test during inference
            post_process='default', # Use 'default' post-processing approach.
            shift_heatmap=True, # Shift and align the flipped heatmap to achieve
↳higher performance
            modulate_kernel=11)) # Gaussian kernel size for modulation. Only used for
↳"post_process='unbiased'"

data_cfg = dict(
    image_size=[192, 256], # Size of model input resolution
    heatmap_size=[48, 64], # Size of the output heatmap
    num_output_channels=channel_cfg['num_output_channels'], # Number of output
↳channels
    num_joints=channel_cfg['dataset_joints'], # Number of joints
    dataset_channel=channel_cfg['dataset_channel'], # Dataset supported channels
    inference_channel=channel_cfg['inference_channel'], # Channels to output
    soft_nms=False, # Whether to perform soft-nms during inference
    nms_thr=1.0, # Threshold for non maximum suppression.
    oks_thr=0.9, # Threshold of oks (object keypoint similarity) score during nms
    vis_thr=0.2, # Threshold of keypoint visibility
    use_gt_bbox=False, # Whether to use ground-truth bounding box during testing
    det_bbox_thr=0.0, # Threshold of detected bounding box score. Used when 'use_gt_
↳bbox=True'
    bbox_file='data/coco/person_detection_results/' # Path to the bounding box
↳detection file
    'COCO_val2017_detections_AP_H_56_person.json',
)

train_pipeline = [
    dict(type='LoadImageFromFile'), # Loading image from file
    dict(type='TopDownRandomFlip', # Perform random flip augmentation
        flip_prob=0.5), # Probability of implementing flip
    dict(
        type='TopDownHalfBodyTransform', # Config of TopDownHalfBodyTransform data-
↳augmentation
        num_joints_half_body=8, # Threshold of performing half-body transform.
        prob_half_body=0.3), # Probability of implementing half-body transform
    dict(
        type='TopDownGetRandomScaleRotation', # Config of
↳TopDownGetRandomScaleRotation
        rot_factor=40, # Rotating to ``[-2*rot_factor, 2*rot_factor]``.
        scale_factor=0.5), # Scaling to ``[1-scale_factor, 1+scale_factor]``.
    dict(type='TopDownAffine', # Affine transform the image to make input.
        use_udp=False), # Do not use unbiased data processing.

```

(continues on next page)

(continued from previous page)

```

dict(type='ToTensor'), # Convert other types to tensor type pipeline
dict(
    type='NormalizeTensor', # Normalize input tensors
    mean=[0.485, 0.456, 0.406], # Mean values of different channels to
↪normalize
    std=[0.229, 0.224, 0.225]), # Std values of different channels to normalize
dict(type='TopDownGenerateTarget', # Generate heatmap target. Different
↪encoding types supported.
    sigma=2), # Sigma of heatmap gaussian
dict(
    type='Collect', # Collect pipeline that decides which keys in the data
↪should be passed to the detector
    keys=['img', 'target', 'target_weight'], # Keys of input
    meta_keys=[ # Meta keys of input
        'image_file', 'joints_3d', 'joints_3d_visible', 'center', 'scale',
        'rotation', 'bbox_score', 'flip_pairs'
    ]),
]

val_pipeline = [
    dict(type='LoadImageFromFile'), # Loading image from file
    dict(type='TopDownAffine'), # Affine transform the image to make input.
    dict(type='ToTensor'), # Config of ToTensor
    dict(
        type='NormalizeTensor',
        mean=[0.485, 0.456, 0.406], # Mean values of different channels to
↪normalize
        std=[0.229, 0.224, 0.225]), # Std values of different channels to normalize
    dict(
        type='Collect', # Collect pipeline that decides which keys in the data
↪should be passed to the detector
        keys=['img'], # Keys of input
        meta_keys=[ # Meta keys of input
            'image_file', 'center', 'scale', 'rotation', 'bbox_score',
            'flip_pairs'
        ]),
]

test_pipeline = val_pipeline

data_root = 'data/coco' # Root of the dataset
data = dict( # Config of data
    samples_per_gpu=64, # Batch size of each single GPU during training
    workers_per_gpu=2, # Workers to pre-fetch data for each single GPU
    val_dataloader=dict(samples_per_gpu=32), # Batch size of each single GPU
↪during validation
    test_dataloader=dict(samples_per_gpu=32), # Batch size of each single GPU
↪during testing
    train=dict( # Training dataset config
        type='TopDownCocoDataset', # Name of dataset
        ann_file=f'{data_root}/annotations/person_keypoints_train2017.json', #
↪Path to annotation file

```

(continues on next page)

(continued from previous page)

```

        img_prefix=f'{data_root}/train2017/',
        data_cfg=data_cfg,
        pipeline=train_pipeline),
    val=dict( # Validation dataset config
        type='TopDownCocoDataset', # Name of dataset
        ann_file=f'{data_root}/annotations/person_keypoints_val2017.json', # Path
        ↪to annotation file
        img_prefix=f'{data_root}/val2017/',
        data_cfg=data_cfg,
        pipeline=val_pipeline),
    test=dict( # Testing dataset config
        type='TopDownCocoDataset', # Name of dataset
        ann_file=f'{data_root}/annotations/person_keypoints_val2017.json', # Path
        ↪to annotation file
        img_prefix=f'{data_root}/val2017/',
        data_cfg=data_cfg,
        pipeline=val_pipeline),
)

```

35.3 FAQ

35.3.1 Use intermediate variables in configs

Some intermediate variables are used in the config files, like `train_pipeline/val_pipeline/test_pipeline` etc.

For Example, we would like to first define `train_pipeline/val_pipeline/test_pipeline` and pass them into data. Thus, `train_pipeline/val_pipeline/test_pipeline` are intermediate variable.

TUTORIAL 1: FINETUNING MODELS

Detectors pre-trained on the COCO dataset can serve as a good pre-trained model for other datasets, e.g., COCO-WholeBody Dataset. This tutorial provides instruction for users to use the models provided in the [Model Zoo](#) for other datasets to obtain better performance.

- *Outline*
- *Modify Head*
- *Modify Dataset*
- *Modify Training Schedule*
- *Use Pre-Trained Model*

36.1 Outline

There are two steps to finetune a model on a new dataset.

- Add support for the new dataset following *Tutorial 2: Adding New Dataset*.
- Modify the configs as will be discussed in this tutorial.

To finetune on the custom datasets, the users need to modify four parts in the config.

36.2 Modify Head

Then the new config needs to modify the model according to the keypoint numbers of the new datasets. By only changing `out_channels` in the `keypoint_head`. For example, we have 133 keypoints for COCO-WholeBody, and we have 17 keypoints for COCO.

```
channel_cfg = dict(
    num_output_channels=133, # changing from 17 to 133
    dataset_joints=133, # changing from 17 to 133
    dataset_channel=[
        list(range(133)), # changing from 17 to 133
    ],
    inference_channel=list(range(133))) # changing from 17 to 133

# model settings
model = dict(
    type='TopDown',
```

(continues on next page)

(continued from previous page)

```

pretrained='https://download.openmmlab.com/mmpose/'
'pretrain_models/hrnet_w48-8ef0771d.pth',
backbone=dict(
    type='HRNet',
    in_channels=3,
    extra=dict(
        stage1=dict(
            num_modules=1,
            num_branches=1,
            block='BOTTLENECK',
            num_blocks=(4, ),
            num_channels=(64, )),
        stage2=dict(
            num_modules=1,
            num_branches=2,
            block='BASIC',
            num_blocks=(4, 4),
            num_channels=(48, 96)),
        stage3=dict(
            num_modules=4,
            num_branches=3,
            block='BASIC',
            num_blocks=(4, 4, 4),
            num_channels=(48, 96, 192)),
        stage4=dict(
            num_modules=3,
            num_branches=4,
            block='BASIC',
            num_blocks=(4, 4, 4, 4),
            num_channels=(48, 96, 192, 384))),
    ),
keypoint_head=dict(
    type='TopdownHeatmapSimpleHead',
    in_channels=48,
    out_channels=channel_cfg['num_output_channels'], # modify this
    num_deconv_layers=0,
    extra=dict(final_conv_kernel=1, ),
    loss_keypoint=dict(type='JointsMSELoss', use_target_weight=True)),
train_cfg=dict(),
test_cfg=dict(
    flip_test=True,
    post_process='unbiased',
    shift_heatmap=True,
    modulate_kernel=17))

```

Note that the pretrained='https://download.openmmlab.com/mmpose/pretrain_models/hrnet_w48-8ef0771d.pth' setting is used for initializing backbone. If you are training a new model from ImageNet-pretrained weights, this is for you. However, this setting is not related to our task at hand. What we need is load_from, which will be discussed later.

36.3 Modify dataset

The users may also need to prepare the dataset and write the configs about dataset. MMPose supports multiple (10+) dataset, including COCO, COCO-WholeBody and MPII-TRB.

```
data_root = 'data/coco'
data = dict(
    samples_per_gpu=32,
    workers_per_gpu=2,
    val_dataloader=dict(samples_per_gpu=32),
    test_dataloader=dict(samples_per_gpu=32),
    train=dict(
        type='TopDownCocoWholeBodyDataset', # modify the name of the dataset
        ann_file=f'{data_root}/annotations/coco_wholebody_train_v1.0.json', # modify the
↪path to the annotation file
        img_prefix=f'{data_root}/train2017/',
        data_cfg=data_cfg,
        pipeline=train_pipeline),
    val=dict(
        type='TopDownCocoWholeBodyDataset', # modify the name of the dataset
        ann_file=f'{data_root}/annotations/coco_wholebody_val_v1.0.json', # modify the
↪path to the annotation file
        img_prefix=f'{data_root}/val2017/',
        data_cfg=data_cfg,
        pipeline=val_pipeline),
    test=dict(
        type='TopDownCocoWholeBodyDataset', # modify the name of the dataset
        ann_file=f'{data_root}/annotations/coco_wholebody_val_v1.0.json', # modify the
↪path to the annotation file
        img_prefix=f'{data_root}/val2017/',
        data_cfg=data_cfg,
        pipeline=val_pipeline)
)
```

36.4 Modify training schedule

The finetuning hyperparameters vary from the default schedule. It usually requires smaller learning rate and less training epochs

```
optimizer = dict(
    type='Adam',
    lr=5e-4, # reduce it
)
optimizer_config = dict(grad_clip=None)
# learning policy
lr_config = dict(
    policy='step',
    warmup='linear',
    warmup_iters=500,
    warmup_ratio=0.001,
```

(continues on next page)

(continued from previous page)

```
    step=[170, 200]) # reduce it
total_epochs = 210 # reduce it
```

36.5 Use pre-trained model

Users can load a pre-trained model by setting the `load_from` field of the config to the model's path or link. The users might need to download the model weights before training to avoid the download time during training.

```
# use the pre-trained model for the whole HRNet
load_from = 'https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_384x288_
↳dark-741844ba_20200812.pth' # model path can be found in model zoo
```

TUTORIAL 2: ADDING NEW DATASET

37.1 Customize datasets by reorganizing data to COCO format

The simplest way to use the custom dataset is to convert your annotation format to COCO dataset format.

The annotation json files in COCO format has the following necessary keys:

```
'images': [  
  {  
    'file_name': '0000000001268.jpg',  
    'height': 427,  
    'width': 640,  
    'id': 1268  
  },  
  ...  
],  
'annotations': [  
  {  
    'segmentation': [[426.36,  
      ...  
      424.34,  
      223.3]],  
    'keypoints': [0,0,0,  
      0,0,0,  
      0,0,0,  
      427,220,2,  
      443,222,2,  
      414,228,2,  
      449,232,2,  
      408,248,1,  
      454,261,2,  
      0,0,0,  
      0,0,0,  
      411,287,2,  
      431,287,2,  
      0,0,0,  
      458,265,2,  
      0,0,0,  
      466,300,1],  
    'num_keypoints': 10,  
    'area': 3894.5826,  
  },  
  ...  
]
```

(continues on next page)

(continued from previous page)

```

        'iscrowd': 0,
        'image_id': 1268,
        'bbox': [402.34, 205.02, 65.26, 88.45],
        'category_id': 1,
        'id': 215218
    },
    ...
],
'categories': [
    {'id': 1, 'name': 'person'},
]

```

There are three necessary keys in the json file:

- **images**: contains a list of images with their information like `file_name`, `height`, `width`, and `id`.
- **annotations**: contains the list of instance annotations.
- **categories**: contains the category name ('person') and its ID (1).

37.2 Create a custom dataset_info config file for the dataset

Add a new dataset info config file.

```
configs/_base_/datasets/custom.py
```

An example of the dataset config is as follows.

`keypoint_info` contains the information about each keypoint.

1. **name**: the keypoint name. The keypoint name must be unique.
2. **id**: the keypoint id.
3. **color**: ([B, G, R]) is used for keypoint visualization.
4. **type**: 'upper' or 'lower', will be used in data augmentation.
5. **swap**: indicates the 'swap pair' (also known as 'flip pair'). When applying image horizontal flip, the left part will become the right part. We need to flip the keypoints accordingly.

`skeleton_info` contains the information about the keypoint connectivity, which is used for visualization.

`joint_weights` assigns different loss weights to different keypoints.

`sigmas` is used to calculate the OKS score. Please read [keypoints-eval](#) to learn more about it.

```

dataset_info = dict(
    dataset_name='coco',
    paper_info=dict(
        author='Lin, Tsung-Yi and Maire, Michael and '
        'Belongie, Serge and Hays, James and '
        'Perona, Pietro and Ramanan, Deva and '
        r'Doll{\'a}r, Piotr and Zitnick, C Lawrence',
        title='Microsoft coco: Common objects in context',
        container='European conference on computer vision',
        year='2014',

```

(continues on next page)

(continued from previous page)

```

    homepage='http://cocodataset.org/',
),
keypoint_info={
    0:
    dict(name='nose', id=0, color=[51, 153, 255], type='upper', swap=''),
    1:
    dict(
        name='left_eye',
        id=1,
        color=[51, 153, 255],
        type='upper',
        swap='right_eye'),
    2:
    dict(
        name='right_eye',
        id=2,
        color=[51, 153, 255],
        type='upper',
        swap='left_eye'),
    3:
    dict(
        name='left_ear',
        id=3,
        color=[51, 153, 255],
        type='upper',
        swap='right_ear'),
    4:
    dict(
        name='right_ear',
        id=4,
        color=[51, 153, 255],
        type='upper',
        swap='left_ear'),
    5:
    dict(
        name='left_shoulder',
        id=5,
        color=[0, 255, 0],
        type='upper',
        swap='right_shoulder'),
    6:
    dict(
        name='right_shoulder',
        id=6,
        color=[255, 128, 0],
        type='upper',
        swap='left_shoulder'),
    7:
    dict(
        name='left_elbow',
        id=7,
        color=[0, 255, 0],

```

(continues on next page)

(continued from previous page)

```
        type='upper',
        swap='right_elbow'),
8:
dict(
    name='right_elbow',
    id=8,
    color=[255, 128, 0],
    type='upper',
    swap='left_elbow'),
9:
dict(
    name='left_wrist',
    id=9,
    color=[0, 255, 0],
    type='upper',
    swap='right_wrist'),
10:
dict(
    name='right_wrist',
    id=10,
    color=[255, 128, 0],
    type='upper',
    swap='left_wrist'),
11:
dict(
    name='left_hip',
    id=11,
    color=[0, 255, 0],
    type='lower',
    swap='right_hip'),
12:
dict(
    name='right_hip',
    id=12,
    color=[255, 128, 0],
    type='lower',
    swap='left_hip'),
13:
dict(
    name='left_knee',
    id=13,
    color=[0, 255, 0],
    type='lower',
    swap='right_knee'),
14:
dict(
    name='right_knee',
    id=14,
    color=[255, 128, 0],
    type='lower',
    swap='left_knee'),
15:
```

(continues on next page)

(continued from previous page)

```

dict(
    name='left_ankle',
    id=15,
    color=[0, 255, 0],
    type='lower',
    swap='right_ankle'),
16:
dict(
    name='right_ankle',
    id=16,
    color=[255, 128, 0],
    type='lower',
    swap='left_ankle')
},
skeleton_info={
    0:
dict(link=('left_ankle', 'left_knee'), id=0, color=[0, 255, 0]),
    1:
dict(link=('left_knee', 'left_hip'), id=1, color=[0, 255, 0]),
    2:
dict(link=('right_ankle', 'right_knee'), id=2, color=[255, 128, 0]),
    3:
dict(link=('right_knee', 'right_hip'), id=3, color=[255, 128, 0]),
    4:
dict(link=('left_hip', 'right_hip'), id=4, color=[51, 153, 255]),
    5:
dict(link=('left_shoulder', 'left_hip'), id=5, color=[51, 153, 255]),
    6:
dict(link=('right_shoulder', 'right_hip'), id=6, color=[51, 153, 255]),
    7:
dict(
    link=('left_shoulder', 'right_shoulder'),
    id=7,
    color=[51, 153, 255]),
    8:
dict(link=('left_shoulder', 'left_elbow'), id=8, color=[0, 255, 0]),
    9:
dict(
    link=('right_shoulder', 'right_elbow'), id=9, color=[255, 128, 0]),
    10:
dict(link=('left_elbow', 'left_wrist'), id=10, color=[0, 255, 0]),
    11:
dict(link=('right_elbow', 'right_wrist'), id=11, color=[255, 128, 0]),
    12:
dict(link=('left_eye', 'right_eye'), id=12, color=[51, 153, 255]),
    13:
dict(link=('nose', 'left_eye'), id=13, color=[51, 153, 255]),
    14:
dict(link=('nose', 'right_eye'), id=14, color=[51, 153, 255]),
    15:
dict(link=('left_eye', 'left_ear'), id=15, color=[51, 153, 255]),
    16:

```

(continues on next page)

(continued from previous page)

```

dict(link=('right_eye', 'right_ear'), id=16, color=[51, 153, 255]),
17:
dict(link=('left_ear', 'left_shoulder'), id=17, color=[51, 153, 255]),
18:
dict(
    link=('right_ear', 'right_shoulder'), id=18, color=[51, 153, 255])
},
joint_weights=[
    1., 1., 1., 1., 1., 1., 1., 1.2, 1.2, 1.5, 1.5, 1., 1., 1.2, 1.2, 1.5,
    1.5
],
sigmas=[
    0.026, 0.025, 0.025, 0.035, 0.035, 0.079, 0.079, 0.072, 0.072, 0.062,
    0.062, 0.107, 0.107, 0.087, 0.087, 0.089, 0.089
]
])

```

37.3 Create a custom dataset class

1. First create a package inside the mmpose/datasets/datasets folder.
2. Create a class definition of your dataset in the package folder and register it in the registry with a name. Without a name, it will keep giving the error. `KeyError: 'XXXXX is not in the dataset registry'`

```

@DATASETS.register_module(name='MyCustomDataset')
class MyCustomDataset(SomeOtherBaseClassAsPerYourNeed):

```

3. Make sure you have updated the `__init__.py` of your package folder
4. Make sure you have updated the `__init__.py` of the dataset package folder.

37.4 Create a custom training config file

Create a custom training config file as per your need and the model/architecture you want to use in the configs folder. You may modify an existing config file to use the new custom dataset.

In configs/my_custom_config.py:

```

...
# dataset settings
dataset_type = 'MyCustomDataset'
...
data = dict(
    samples_per_gpu=2,
    workers_per_gpu=2,
    train=dict(
        type=dataset_type,
        ann_file='path/to/your/train/json',
        img_prefix='path/to/your/train/img',
        ...),
    val=dict(

```

(continues on next page)

(continued from previous page)

```
    type=dataset_type,
    ann_file='path/to/your/val/json',
    img_prefix='path/to/your/val/img',
    ...),
    test=dict(
        type=dataset_type,
        ann_file='path/to/your/test/json',
        img_prefix='path/to/your/test/img',
        ...))
    ...
```

Make sure you have provided all the paths correctly.

TUTORIAL 3: CUSTOM DATA PIPELINES

38.1 Design of Data pipelines

Following typical conventions, we use `Dataset` and `DataLoader` for data loading with multiple workers. `Dataset` returns a dict of data items corresponding the arguments of models' forward method. Since the data in pose estimation may not be the same size (image size, gt bbox size, etc.), we introduce a new `DataContainer` type in MMCV to help collect and distribute data of different size. See [here](#) for more details.

The data preparation pipeline and the dataset is decomposed. Usually a dataset defines how to process the annotations and a data pipeline defines all the steps to prepare a data dict. A pipeline consists of a sequence of operations. Each operation takes a dict as input and also output a dict for the next transform.

The operations are categorized into data loading, pre-processing, formatting, label generating.

Here is an pipeline example for Simple Baseline (ResNet50).

```
train_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='TopDownRandomFlip', flip_prob=0.5),
    dict(type='TopDownHalfBodyTransform', num_joints_half_body=8, prob_half_body=0.3),
    dict(type='TopDownGetRandomScaleRotation', rot_factor=40, scale_factor=0.5),
    dict(type='TopDownAffine'),
    dict(type='ToTensor'),
    dict(
        type='NormalizeTensor',
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]),
    dict(type='TopDownGenerateTarget', sigma=2),
    dict(
        type='Collect',
        keys=['img', 'target', 'target_weight'],
        meta_keys=[
            'image_file', 'joints_3d', 'joints_3d_visible', 'center', 'scale',
            'rotation', 'bbox_score', 'flip_pairs'
        ]
    ),
]

val_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='TopDownAffine'),
    dict(type='ToTensor'),
    dict(
```

(continues on next page)

(continued from previous page)

```
        type='NormalizeTensor',
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]),
    dict(
        type='Collect',
        keys=['img'],
        meta_keys=[
            'image_file', 'center', 'scale', 'rotation', 'bbox_score',
            'flip_pairs'
        ]),
]
```

For each operation, we list the related dict fields that are added/updated/removed.

38.1.1 Data loading

LoadImageFromFile

- add: img, img_file

38.1.2 Pre-processing

TopDownRandomFlip

- update: img, joints_3d, joints_3d_visible, center

TopDownHalfBodyTransform

- update: center, scale

TopDownGetRandomScaleRotation

- update: scale, rotation

TopDownAffine

- update: img, joints_3d, joints_3d_visible

NormalizeTensor

- update: img

38.1.3 Generating labels

TopDownGenerateTarget

- add: target, target_weight

38.1.4 Formatting

ToTensor

- update: 'img'

Collect

- add: img_meta (the keys of img_meta is specified by meta_keys)
- remove: all other keys except for those specified by keys

38.2 Extend and use custom pipelines

1. Write a new pipeline in any file, e.g., my_pipeline.py. It takes a dict as input and return a dict.

```
from mmpose.datasets import PIPELINES

@PIPELINES.register_module()
class MyTransform:

    def __call__(self, results):
        results['dummy'] = True
        return results
```

2. Import the new class.

```
from .my_pipeline import MyTransform
```

3. Use it in config files.

```
train_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='TopDownRandomFlip', flip_prob=0.5),
    dict(type='TopDownHalfBodyTransform', num_joints_half_body=8, prob_half_body=0.3),
    dict(type='TopDownGetRandomScaleRotation', rot_factor=40, scale_factor=0.5),
    dict(type='TopDownAffine'),
    dict(type='MyTransform'),
    dict(type='ToTensor'),
    dict(
        type='NormalizeTensor',
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]),
    dict(type='TopDownGenerateTarget', sigma=2),
    dict(
        type='Collect',
        keys=['img', 'target', 'target_weight'],
        meta_keys=[
            'image_file', 'joints_3d', 'joints_3d_visible', 'center', 'scale',
            'rotation', 'bbox_score', 'flip_pairs'
        ]),
]
```


TUTORIAL 4: ADDING NEW MODULES

39.1 Customize optimizer

A customized optimizer could be defined as following. Assume you want to add a optimizer named as `MyOptimizer`, which has arguments `a`, `b`, and `c`. You need to first implement the new optimizer in a file, e.g., in `mmpose/core/optimizer/my_optimizer.py`:

```
from mmcv.runner import OPTIMIZERS
from torch.optim import Optimizer

@OPTIMIZERS.register_module()
class MyOptimizer(Optimizer):

    def __init__(self, a, b, c)
```

Then add this module in `mmpose/core/optimizer/__init__.py` thus the registry will find the new module and add it:

```
from .my_optimizer import MyOptimizer
```

Then you can use `MyOptimizer` in `optimizer` field of config files. In the configs, the optimizers are defined by the field `optimizer` like the following:

```
optimizer = dict(type='SGD', lr=0.02, momentum=0.9, weight_decay=0.0001)
```

To use your own optimizer, the field can be changed as

```
optimizer = dict(type='MyOptimizer', a=a_value, b=b_value, c=c_value)
```

We already support to use all the optimizers implemented by PyTorch, and the only modification is to change the `optimizer` field of config files. For example, if you want to use ADAM, though the performance will drop a lot, the modification could be as the following.

```
optimizer = dict(type='Adam', lr=0.0003, weight_decay=0.0001)
```

The users can directly set arguments following the [API doc](#) of PyTorch.

39.2 Customize optimizer constructor

Some models may have some parameter-specific settings for optimization, e.g. weight decay for BatchNorm layers. The users can do those fine-grained parameter tuning through customizing optimizer constructor.

```
from mmcv.utils import build_from_cfg

from mmcv.runner import OPTIMIZER_BUILDERS, OPTIMIZERS
from mmpose.utils import get_root_logger
from .cocktail_optimizer import CocktailOptimizer

@OPTIMIZER_BUILDERS.register_module()
class CocktailOptimizerConstructor:

    def __init__(self, optimizer_cfg, paramwise_cfg=None):

    def __call__(self, model):

        return my_optimizer
```

39.2.1 Develop new components

We basically categorize model components into 3 types.

- detectors: the whole pose detector model pipeline, usually contains a backbone and keypoint_head.
- backbone: usually an FCN network to extract feature maps, e.g., ResNet, HRNet.
- keypoint_head: the component for pose estimation task, usually contains some deconv layers.

1. Create a new file `mmpose/models/backbones/my_model.py`.

```
import torch.nn as nn

from ..builder import BACKBONES

@BACKBONES.register_module()
class MyModel(nn.Module):

    def __init__(self, arg1, arg2):
        pass

    def forward(self, x): # should return a tuple
        pass

    def init_weights(self, pretrained=None):
        pass
```

2. Import the module in `mmpose/models/backbones/__init__.py`.

```
from .my_model import MyModel
```

3. Create a new file `mmpose/models/keypoint_heads/my_head.py`.

You can write a new keypoint head inherit from `nn.Module`, and overwrite `init_weights(self)` and `forward(self, x)` method.

```
from ..builder import HEADS

@HEADS.register_module()
class MyHead(nn.Module):

    def __init__(self, arg1, arg2):
        pass

    def forward(self, x):
        pass

    def init_weights(self):
        pass
```

4. Import the module in `mmpose/models/keypoint_heads/__init__.py`

```
from .my_head import MyHead
```

5. Use it in your config file.

For the top-down 2D pose estimation model, we set the module type as `TopDown`.

```
model = dict(
    type='TopDown',
    backbone=dict(
        type='MyModel',
        arg1=xxx,
        arg2=xxx),
    keypoint_head=dict(
        type='MyHead',
        arg1=xxx,
        arg2=xxx))
```

39.2.2 Add new loss

Assume you want to add a new loss as `MyLoss`, for keypoints estimation. To add a new loss function, the users need implement it in `mmpose/models/losses/my_loss.py`. The decorator `weighted_loss` enable the loss to be weighted for each element.

```
import torch
import torch.nn as nn

from mmpose.models import LOSSES

def my_loss(pred, target):
    assert pred.size() == target.size() and target.numel() > 0
    loss = torch.abs(pred - target)
    loss = torch.mean(loss)
    return loss
```

(continues on next page)

(continued from previous page)

```

@LOSSES.register_module()
class MyLoss(nn.Module):

    def __init__(self, use_target_weight=False):
        super(MyLoss, self).__init__()
        self.criterion = my_loss()
        self.use_target_weight = use_target_weight

    def forward(self, output, target, target_weight):
        batch_size = output.size(0)
        num_joints = output.size(1)

        heatmaps_pred = output.reshape(
            (batch_size, num_joints, -1)).split(1, 1)
        heatmaps_gt = target.reshape((batch_size, num_joints, -1)).split(1, 1)

        loss = 0.

        for idx in range(num_joints):
            heatmap_pred = heatmaps_pred[idx].squeeze(1)
            heatmap_gt = heatmaps_gt[idx].squeeze(1)
            if self.use_target_weight:
                loss += self.criterion(
                    heatmap_pred * target_weight[:, idx],
                    heatmap_gt * target_weight[:, idx])
            else:
                loss += self.criterion(heatmap_pred, heatmap_gt)

        return loss / num_joints

```

Then the users need to add it in the `mmpose/models/losses/__init__.py`.

```

from .my_loss import MyLoss, my_loss

```

To use it, modify the `loss_keypoint` field in the model.

```

loss_keypoint=dict(type='MyLoss', use_target_weight=False)

```

TUTORIAL 5: EXPORTING A MODEL TO ONNX

Open Neural Network Exchange ([ONNX](#)) is an open ecosystem that empowers AI developers to choose the right tools as their project evolves.

- *Supported Models*
- *Usage*
 - *Prerequisite*

40.1 Supported Models

So far, our codebase supports onnx exporting from pytorch models trained with MMPose. The supported models include:

- ResNet
- HRNet
- HigherHRNet

40.2 Usage

For simple exporting, you can use the [script](#) here. Note that the package `onnx` and `onnxruntime` are required for verification after exporting.

40.2.1 Prerequisite

First, install onnx.

```
pip install onnx onnxruntime
```

We provide a python script to export the pytorch model trained by MMPose to ONNX.

```
python tools/deployment/pytorch2onnx.py ${CONFIG_FILE} ${CHECKPOINT_FILE} [--shape $  
↪ ${SHAPE}] \  
    [--verify] [--show] [--output-file ${OUTPUT_FILE}] [--opset-version ${VERSION}]
```

Optional arguments:

- `--shape`: The shape of input tensor to the model. If not specified, it will be set to 1 3 256 192.

- `--verify`: Determines whether to verify the exported model, runnably and numerically. If not specified, it will be set to `False`.
- `--show`: Determines whether to print the architecture of the exported model. If not specified, it will be set to `False`.
- `--output-file`: The output onnx model name. If not specified, it will be set to `tmp.onnx`.
- `--opset-version`: Determines the operation set version of onnx, we recommend you to use a higher version such as 11 for compatibility. If not specified, it will be set to 11.

Please fire an issue if you discover any checkpoints that are not perfectly exported or suffer some loss in accuracy.

TUTORIAL 6: CUSTOMIZE RUNTIME SETTINGS

In this tutorial, we will introduce some methods about how to customize optimization methods, training schedules, workflow and hooks when running your own settings for the project.

- *Customize Optimization Methods*
 - *Customize optimizer supported by PyTorch*
 - *Customize self-implemented optimizer*
 - * *1. Define a new optimizer*
 - * *2. Add the optimizer to registry*
 - * *3. Specify the optimizer in the config file*
 - *Customize optimizer constructor*
 - *Additional settings*
- *Customize Training Schedules*
- *Customize Workflow*
- *Customize Hooks*
 - *Customize self-implemented hooks*
 - * *1. Implement a new hook*
 - * *2. Register the new hook*
 - * *3. Modify the config*
 - *Use hooks implemented in MMCV*
 - *Modify default runtime hooks*
 - * *Checkpoint config*
 - * *Log config*
 - * *Evaluation config*

41.1 Customize Optimization Methods

41.1.1 Customize optimizer supported by PyTorch

We already support to use all the optimizers implemented by PyTorch, and the only modification is to change the `optimizer` field of config files. For example, if you want to use Adam, the modification could be as the following.

```
optimizer = dict(type='Adam', lr=0.0003, weight_decay=0.0001)
```

To modify the learning rate of the model, the users only need to modify the `lr` in the config of optimizer. The users can directly set arguments following the [API doc](#) of PyTorch.

For example, if you want to use Adam with the setting like `torch.optim.Adam(params, lr=0.001, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False)` in PyTorch, the modification could be set as the following.

```
optimizer = dict(type='Adam', lr=0.001, betas=(0.9, 0.999), eps=1e-08, weight_decay=0,
    ↪amsgrad=False)
```

41.1.2 Customize self-implemented optimizer

1. Define a new optimizer

A customized optimizer could be defined as following.

Assume you want to add an optimizer named `MyOptimizer`, which has arguments `a`, `b`, and `c`. You need to create a new directory named `mmpose/core/optimizer`. And then implement the new optimizer in a file, e.g., in `mmpose/core/optimizer/my_optimizer.py`:

```
from .builder import OPTIMIZERS
from torch.optim import Optimizer

@OPTIMIZERS.register_module()
class MyOptimizer(Optimizer):

    def __init__(self, a, b, c):
```

2. Add the optimizer to registry

To find the above module defined above, this module should be imported into the main namespace at first. There are two ways to achieve it.

- Modify `mmpose/core/optimizer/__init__.py` to import it.

The newly defined module should be imported in `mmpose/core/optimizer/__init__.py` so that the registry will find the new module and add it:

```
from .my_optimizer import MyOptimizer
```

- Use `custom_imports` in the config to manually import it

```
custom_imports = dict(imports=['mmpose.core.optimizers.my_optimizer'], allow_failed_
↳ imports=False)
```

The module `mmpose.core.optimizer.my_optimizer` will be imported at the beginning of the program and the class `MyOptimizer` is then automatically registered. Note that only the package containing the class `MyOptimizer` should be imported. `mmpose.core.optimizer.my_optimizer.MyOptimizer` **cannot** be imported directly.

3. Specify the optimizer in the config file

Then you can use `MyOptimizer` in `optimizer` field of config files. In the configs, the optimizers are defined by the field `optimizer` like the following:

```
optimizer = dict(type='SGD', lr=0.02, momentum=0.9, weight_decay=0.0001)
```

To use your own optimizer, the field can be changed to

```
optimizer = dict(type='MyOptimizer', a=a_value, b=b_value, c=c_value)
```

41.1.3 Customize optimizer constructor

Some models may have some parameter-specific settings for optimization, e.g. weight decay for BatchNorm layers. The users can do those fine-grained parameter tuning through customizing optimizer constructor.

```
from mmcv.utils import build_from_cfg

from mmcv.runner.optimizer import OPTIMIZER_BUILDERS, OPTIMIZERS
from mmpose.utils import get_root_logger
from .my_optimizer import MyOptimizer

@OPTIMIZER_BUILDERS.register_module()
class MyOptimizerConstructor:

    def __init__(self, optimizer_cfg, paramwise_cfg=None):
        pass

    def __call__(self, model):

        return my_optimizer
```

The default optimizer constructor is implemented [here](#), which could also serve as a template for new optimizer constructor.

41.1.4 Additional settings

Tricks not implemented by the optimizer should be implemented through optimizer constructor (e.g., set parameter-wise learning rates) or hooks. We list some common settings that could stabilize the training or accelerate the training. Feel free to create PR, issue for more settings.

- **Use gradient clip to stabilize training:** Some models need gradient clip to clip the gradients to stabilize the training process. An example is as below:

```
optimizer_config = dict(grad_clip=dict(max_norm=35, norm_type=2))
```

- **Use momentum schedule to accelerate model convergence:** We support momentum scheduler to modify model's momentum according to learning rate, which could make the model converge in a faster way. Momentum scheduler is usually used with LR scheduler, for example, the following config is used in 3D detection to accelerate convergence. For more details, please refer to the implementation of [CyclicLrUpdater](#) and [CyclicMomentumUpdater](#).

```
lr_config = dict(
    policy='cyclic',
    target_ratio=(10, 1e-4),
    cyclic_times=1,
    step_ratio_up=0.4,
)
momentum_config = dict(
    policy='cyclic',
    target_ratio=(0.85 / 0.95, 1),
    cyclic_times=1,
    step_ratio_up=0.4,
)
```

41.2 Customize Training Schedules

we use step learning rate with default value in config files, this calls [StepLRHook](#) in MMCV. We support many other learning rate schedule [here](#), such as [CosineAnnealing](#) and [Poly](#) schedule. Here are some examples

- Poly schedule:

```
lr_config = dict(policy='poly', power=0.9, min_lr=1e-4, by_epoch=False)
```

- CosineAnnealing schedule:

```
lr_config = dict(
    policy='CosineAnnealing',
    warmup='linear',
    warmup_iters=1000,
    warmup_ratio=1.0 / 10,
    min_lr_ratio=1e-5)
```

41.3 Customize Workflow

By default, we recommend users to use `EpochEvalHook` to do evaluation after training epoch, but they can still use `val` workflow as an alternative.

Workflow is a list of (phase, epochs) to specify the running order and epochs. By default it is set to be

```
workflow = [('train', 1)]
```

which means running 1 epoch for training. Sometimes user may want to check some metrics (e.g. loss, accuracy) about the model on the validate set. In such case, we can set the workflow as

```
[('train', 1), ('val', 1)]
```

so that 1 epoch for training and 1 epoch for validation will be run iteratively.

Note:

1. The parameters of model will not be updated during val epoch.
2. Keyword `total_epochs` in the config only controls the number of training epochs and will not affect the validation workflow.
3. Workflows `[('train', 1), ('val', 1)]` and `[('train', 1)]` will not change the behavior of `EpochEvalHook` because `EpochEvalHook` is called by `after_train_epoch` and validation workflow only affect hooks that are called through `after_val_epoch`. Therefore, the only difference between `[('train', 1), ('val', 1)]` and `[('train', 1)]` is that the runner will calculate losses on validation set after each training epoch.

41.4 Customize Hooks

41.4.1 Customize self-implemented hooks

1. Implement a new hook

Here we give an example of creating a new hook in MMPose and using it in training.

```
from mmcv.runner import HOOKS, Hook

@HOOKS.register_module()
class MyHook(Hook):

    def __init__(self, a, b):
        pass

    def before_run(self, runner):
        pass

    def after_run(self, runner):
        pass
```

(continues on next page)

(continued from previous page)

```
def before_epoch(self, runner):  
    pass  
  
def after_epoch(self, runner):  
    pass  
  
def before_iter(self, runner):  
    pass  
  
def after_iter(self, runner):  
    pass
```

Depending on the functionality of the hook, the users need to specify what the hook will do at each stage of the training in `before_run`, `after_run`, `before_epoch`, `after_epoch`, `before_iter`, and `after_iter`.

2. Register the new hook

Then we need to make `MyHook` imported. Assuming the file is in `mmpose/core/utils/my_hook.py` there are two ways to do that:

- Modify `mmpose/core/utils/__init__.py` to import it.

The newly defined module should be imported in `mmpose/core/utils/__init__.py` so that the registry will find the new module and add it:

```
from .my_hook import MyHook
```

- Use `custom_imports` in the config to manually import it

```
custom_imports = dict(imports=['mmpose.core.utils.my_hook'], allow_failed_imports=False)
```

3. Modify the config

```
custom_hooks = [  
    dict(type='MyHook', a=a_value, b=b_value)  
]
```

You can also set the priority of the hook by adding key `priority` to `'NORMAL'` or `'HIGHEST'` as below

```
custom_hooks = [  
    dict(type='MyHook', a=a_value, b=b_value, priority='NORMAL')  
]
```

By default the hook's priority is set as `NORMAL` during registration.

41.4.2 Use hooks implemented in MMCV

If the hook is already implemented in MMCV, you can directly modify the config to use the hook as below

```
mmcv_hooks = [
    dict(type='MMCVHook', a=a_value, b=b_value, priority='NORMAL')
]
```

41.4.3 Modify default runtime hooks

There are some common hooks that are not registered through `custom_hooks` but has been registered by default when importing MMCV, they are

- `log_config`
- `checkpoint_config`
- `evaluation`
- `lr_config`
- `optimizer_config`
- `momentum_config`

In those hooks, only the logger hook has the `VERY_LOW` priority, others' priority are `NORMAL`. The above-mentioned tutorials already cover how to modify `optimizer_config`, `momentum_config`, and `lr_config`. Here we reveals how what we can do with `log_config`, `checkpoint_config`, and `evaluation`.

Checkpoint config

The MMCV runner will use `checkpoint_config` to initialize `CheckpointHook`.

```
checkpoint_config = dict(interval=1)
```

The users could set `max_keep_ckpts` to only save only small number of checkpoints or decide whether to store state dict of optimizer by `save_optimizer`. More details of the arguments are [here](#)

Log config

The `log_config` wraps multiple logger hooks and enables to set intervals. Now MMCV supports `WandbLoggerHook`, `MlflowLoggerHook`, and `TensorboardLoggerHook`. The detail usages can be found in the [doc](#).

```
log_config = dict(
    interval=50,
    hooks=[
        dict(type='TextLoggerHook'),
        dict(type='TensorboardLoggerHook')
    ])

```

Evaluation config

The config of `evaluation` will be used to initialize the `EvalHook`. Except the key `interval`, other arguments such as `metric` will be passed to the `dataset.evaluate()`

```
evaluation = dict(interval=1, metric='mAP')
```


TUTORIAL 7DEVELOP APPLICATIONS WITH WEBCAM API

MMPose Webcam API is a toolkit to develop pose-empowered applications. This tutorial introduces the features and usage of Webcam API. More technical details can be found at [API Reference](#).

- *Overview*
- *An Example of Webcam Applications*
 - *Run the demo*
 - *Configs*
 - * *Buffer configurations*
 - * *Hot-key configurations*
 - *Architecture of a webcam application*
- *Extending Webcam API with Custom Nodes*
 - *Custom nodes for general functions*
 - * *Inherit from Node class*
 - * *Implement __init__() method*
 - * *Implement process() method*
 - * *Implement bypass() method*
 - *Custom nodes for visualization*
 - * *Inherit from BaseVisualizerNode class*
 - * *Implement draw() method*

42.1 Overview

Webcam API is composed of the following main modules (Shown in Fig. 1):

1. **WebcamExecutor** (See [webcam_executor.py](#)): The interface to build and launch the application program, and perform video capturing and displaying. Besides, WebcamExecutor builds a certain number of functional modules according to the config to perform different basic functions like model inference, data processing, logical decision, and image drawing. when launched, the WebcamExecutor continually reads video frames, controls the data flow among all function modules, and finally displays the processed results. And below are concepts related to WebcamExecutor:
 1. **Config** : The configuration file contains the parameters of the **WebcamExecutor** and all function modules. Webcam API uses python files as configs, following the common practice of OpenMMLab;

2. **Launcher** (e.g. `webcam_demo.py`): A script to load the config file, build `WebcamExecutor` and invoke its `run()` method to start the application program;
2. **Node** (See `node.py`): The interface of function module. One node usually implements a basic function. For example, `DetectorNode` performs object detection from the frame; `ObjectVisualizerNode` draws the bbox and keypoints of objects; `RecorderNode` write the frames into a local video file. Users can also add custom nodes by inheriting the `Node` interface.
3. **Utils**: Utility modules and functions including:
 1. **Message** (See `message.py`): The data interface of the `WebcamExecutor` and `Node`. Message instances may contain images, model inference results, text information, or arbitrary custom data;
 2. **Buffer** (See `buffer.py`): The container of `Message` instances for asynchronous communication between nodes. A node fetches the input from its input buffers once it's ready, and put the output into its output buffer;
 3. **Event** (See `event.py`): The event manager supports event communication within the program. Different from the data message that follows a route defined by the config, an event can be set or responded by the executor or nodes immediately. For example, when the user presses a key on the keyboard, an event will be broadcasted to all nodes. This mechanism is useful in user interaction functions.

42.2 An Example of Webcam Applications

In this section, we will introduce how to build an application by Webcam API via a simple example.

42.2.1 Run the demo

Before we dive into technical details, you can try running this demo first with the following command. What it does is read the video stream from the webcam, display it on the screen and save it to a local file.

```
# python demo/webcam_demo.py --config CONFIG_PATH [--debug]
python demo/webcam_demo.py --config demo/webcam_cfg/test_camera.py
```

42.2.2 Configs

Now let's look at the config used in this demo:

```
executor_cfg = dict(
    name='Test Webcam', # name of the application
    camera_id=0, # camera ID (optionally, it can be a path of an input video file)
    camera_max_fps=30, # maximum FPS to read the video
    nodes=[
        # `MonitorNode` shows the system and application information
        dict(
            type='MonitorNode', # node type
            name='monitor', # node name
            enable_key='m', # hot key to switch on/off
            enable=False, # init status of on/off
            input_buffer='_frame_', # input buffer
            output_buffer='display'), # output buffer
        # `RecorderNode` saves output to a local file
    ]
)
```

(continues on next page)

(continued from previous page)

```
dict(
    type='RecorderNode', # node name
    name='recorder', # node type
    out_video_file='webcam_output.mp4', # path to save output
    input_buffer='display', # input buffer
    output_buffer='_display_') # output buffer
])
```

As shown above, the content of the config file is a dict named `executor_cfg`, which contains basic parameters (e.g. `name`, `camera_id`, et al. See the [document](#) for details) and node configs (nodes). The node configs are stored in a list, of which each element is a dict that contains parameters of one node. There are 2 nodes in the demo, namely a `DetectorNode` and a `RecorderNode`. See the [document of node](#) for more information.

Buffer configurations

From the demo config, you may have noticed that nodes usually have a special type of parameters: input and output buffers. As noted previously, a buffer is a data container to hold the input and output of nodes. And in the config, we can specify the input and output buffer of each node by buffer names. In the demo config, for example, `MonitorNode` fetches input from a buffer named `"_frame_"`, and puts output to a buffer named `"display"`; and `RecorderNode` fetches input from the buffer `"display"`, and outputs to another buffer `"_display_"`.

In the config, you can assign arbitrary buffer names, and the executor will build buffers accordingly and connect them with the nodes. It's important to note that the following 3 names are reserved for special buffers to exchange data between the executor and nodes:

- `"_input_"`: The buffer to store frames read by the executor for model inference;
- `"_frame_"`: The buffer to store frames read by the executor (same as `"_input_"`) for visualization functions. We use separate inputs for model inference and visualization so they can run asynchronously.
- `"_display_"`: The buffer to store output that has been processed by nodes. The executor will load from this buffer to display.

In an application, the executor will build a **BufferManager** instance to hold all buffers (See [BufferManager document](#) for details).

Hot-key configurations

Some nodes support switch state control by hot-keys. These nodes have the following parameters:

- `enable_key` (str): Specify the hot-key for switch state control;
- `enable` (bool): Set the initial switch state.

The hot-key response is supported by the event mechanism. The executor has a **EventManager** (See [EventManager document](#)) instance to manage all user-defined events in the application. A node can register events at initialization. Registered events can be set, waited, or cleared at run time.

42.2.3 Architecture of a webcam application

Now we have introduced the concept of WebcamExecutor, Node, Buffer, and Event. The architecture of a webcam application can be illustrated as shown in Fig. 2.

42.3 Extending Webcam API with Custom Nodes

Webcam API provides a simple and efficient interface to extend by defining new nodes. In this section, we will show you how to do this via examples.

42.3.1 Custom nodes for general functions

We first introduce the general steps to define new nodes. Here we take `DetectorNode` as an example.

Inherit from Node class

All node classes should inherit from the base class `Node` (See `node.py`) and be registered to the registry `NODES`. So the node instances can be built from configs.

```
from mmpose.apis.webcam.nodes import Node, NODES

@NODES.register_module()
class DetectorNode(Node):
    ...
```

Implement `__init__()` method

The `__init__()` method of `DetectorNode` is implemented as below:

```
def __init__(self,
             name: str,
             model_config: str,
             model_checkpoint: str,
             input_buffer: str,
             output_buffer: Union[str, List[str]],
             enable_key: Optional[Union[str, int]] = None,
             enable: bool = True,
             device: str = 'cuda:0',
             bbox_thr: float = 0.5):

    # Initialize the base class
    super().__init__(name=name, enable_key=enable_key, enable=enable)

    # Initialize parameters
    self.model_config = get_config_path(model_config, 'mmdet')
    self.model_checkpoint = model_checkpoint
    self.device = device.lower()
    self.bbox_thr = bbox_thr
```

(continues on next page)

(continued from previous page)

```

self.model = init_detector(
    self.model_config, self.model_checkpoint, device=self.device)

# Register input/output buffers
self.register_input_buffer(input_buffer, 'input', trigger=True) # Set trigger
self.register_output_buffer(output_buffer)

```

The `__init__()` method usually does the following steps:

1. **Initialize the base class:** Call `super().__init__()` with parameters like `name`, `enable_key` and `enable`;
2. **Initialize node parameters:** In this example, we initialize the parameters like `model_config`, `device`, `bbox_thr` in the node, and load the model with MMDetection APIs.
3. **Register buffers:** A node needs to register its input and output buffers during initialization:
 1. Register each input buffer by `register_input_buffer()` method. This method maps the buffer name (i.e. `input_buffer` from the config) to an indicator (i.e. `"input"` in the example). At runtime, the node can access the data from the registered buffers by indicators (See *Implement process() method*).
 2. Register the output buffers by `register_output_buffer()` method. At runtime, the node output will be stored in every registered output buffer (each buffer will store a deep copy of the node output).

Implement process() method

The `process()` method defines the behavior of a node. In the `DetectorNode` example, we implement detection model inference in the `process()` method:

```

def process(self, input_msgs):

    # Get the input message from the buffer by the indicator 'input'
    input_msg = input_msgs['input']

    # Get image data from the input message
    img = input_msg.get_image()

    # Process model inference using MMDetection API
    preds = inference_detector(self.model, img)
    objects = self._post_process(preds)

    # Assign the detection results into the message
    input_msg.update_objects(objects)

    # Return the message
    return input_msg

```

The `process()` method usually does the following steps:

1. **Get input data:** The argument `input_msgs` contains data fetched from all registered input buffers. Data from a specific buffer can be obtained by the indicator (e.g. `"input"`);
2. **Parse input data:** The input data are usually `FrameMessage` instances (See the [document](#) for details). The node can extract the image data and model inference results from the message;
3. **Process:** In this example, we use MMDetection APIs to detect objects from the input image, and post-process the result format;

4. **Return results:** The detection results are assigned to the `input_msg` by the `update_objects()` method. Then the message is returned by `process()` and will be stored in all registered output buffers to serve as the input of downstream nodes.

Implement `bypass()` method

If a node supports switch state control by hot-keys, its `bypass()` method should be implemented to define the node behavior when turned off. The `bypass()` method has the same function signature as the `process()` method. `DetectorNode` simply outputs the input message in the `bypass()` method as the following:

```
def bypass(self, input_msgs):
    return input_msgs['input']
```

42.3.2 Custom nodes for visualization

Visualizer Node is a special category of nodes for visualization functions. Here we will introduce a simpler interface to extend this kind of nodes. We take `NoticeBoardNode` as an example, whose function is to show text information in the output frames.

Inherit from `BaseVisualizerNode` class

`BaseVisualizerNode` is a subclass of `Node` that partially implements the `process()` method and exposes the `draw()` method as an image editing interface. Visualizer nodes should inherit from `BaseVisualizerNode` and be registered to the registry `NODES`.

```
from mmpose.apis.webcam.nodes import BaseVisualizerNode, NODES

@NODES.register_module()
class NoticeBoardNode(BaseVisualizerNode):
    ...
```

The implementation of `__init__()` in visualizer nodes is similar to it in general nodes. Please refer to `Implement __init__() method`. Note that a visualizer node should register one and only one input buffer with the name "input".

Implement `draw()` method

The `draw()` method has one argument `input_msg`, which is the data fetched from the buffer indicated by "input". The return value of `draw()` is an image in `np.ndarray` type, which will be used to update the image data in `input_msg`. And the updated `input_msg` will be the node output.

We implement the `draw()` method of `NoticeBoardNode` as the following:

```
def draw(self, input_msg: FrameMessage) -> np.ndarray:
    # Get frame image data
    img = input_msg.get_image()

    # Create a canvas
    canvas = np.full(img.shape, self.background_color, dtype=img.dtype)

    # Put the text on the canvas image
    x = self.x_offset
```

(continues on next page)

(continued from previous page)

```
y = self.y_offset
max_len = max([len(line) for line in self.content_lines])

def _put_line(line=''):
    nonlocal y
    cv2.putText(canvas, line, (x, y), cv2.FONT_HERSHEY_DUPLEX,
                self.text_scale, self.text_color, 1)
    y += self.y_delta

for line in self.content_lines:
    _put_line(line)

# Copy and paste the valid region of the canvas to the frame image
x1 = max(0, self.x_offset)
x2 = min(img.shape[1], int(x + max_len * self.text_scale * 20))
y1 = max(0, self.y_offset - self.y_delta)
y2 = min(img.shape[0], y)

src1 = canvas[y1:y2, x1:x2]
src2 = img[y1:y2, x1:x2]
img[y1:y2, x1:x2] = cv2.addWeighted(src1, 0.5, src2, 0.5, 0)

# Return the processed image
return img
```

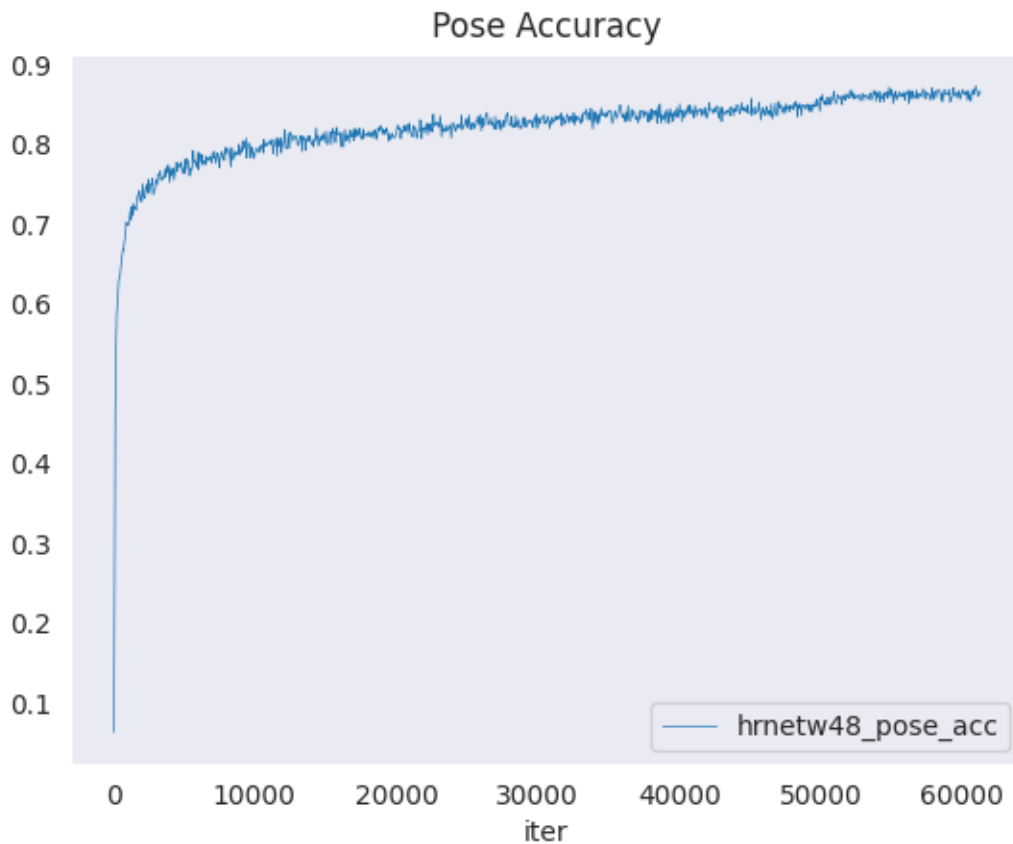

USEFUL TOOLS

Apart from training/testing scripts, We provide lots of useful tools under the `tools/` directory.

- *Log Analysis*
- *Model Complexity (experimental)*
- *Model Conversion*
 - *MMPose model to ONNX (experimental)*
 - *Prepare a model for publishing*
- *Model Serving*
- *Miscellaneous*
 - Evaluating a metric
 - *Print the entire config*

43.1 Log Analysis

`tools/analysis/analyze_logs.py` plots loss/pose acc curves given a training log file. Run `pip install seaborn` first to install the dependency.



```
python tools/analysis/analyze_logs.py plot_curve ${JSON_LOGS} [--keys ${KEYS}] [--title $
↪{TITLE}] [--legend ${LEGEND}] [--backend ${BACKEND}] [--style ${STYLE}] [--out ${OUT_
↪FILE}]
```

Examples:

- Plot the mse loss of some run.

```
python tools/analysis/analyze_logs.py plot_curve log.json --keys loss --legend loss
```

- Plot the acc of some run, and save the figure to a pdf.

```
python tools/analysis/analyze_logs.py plot_curve log.json --keys acc_pose --out_
↪results.pdf
```

- Compare the acc of two runs in the same figure.

```
python tools/analysis/analyze_logs.py plot_curve log1.json log2.json --keys acc_
↪pose --legend run1 run2
```

You can also compute the average training speed.

```
python tools/analysis/analyze_logs.py cal_train_time ${JSON_LOGS} [--include-outliers]
```

- Compute the average training speed for a config file

```
python tools/analysis/analyze_logs.py cal_train_time log.json
```

The output is expected to be like the following.

```
-----Analyze train time of log.json-----
slowest epoch 114, average time is 0.9662
fastest epoch 16, average time is 0.7532
time std over epochs is 0.0426
average iter time: 0.8406 s/iter
```

43.2 Model Complexity (Experimental)

`/tools/analysis/get_flops.py` is a script adapted from [flops-counter.pytorch](#) to compute the FLOPs and params of a given model.

```
python tools/analysis/get_flops.py ${CONFIG_FILE} [--shape ${INPUT_SHAPE}]
```

We will get the result like this

```
=====
Input shape: (1, 3, 256, 192)
Flops: 8.9 GMac
Params: 28.04 M
=====
```

Note: This tool is still experimental and we do not guarantee that the number is absolutely correct.

You may use the result for simple comparisons, but double check it before you adopt it in technical reports or papers.

(1) FLOPs are related to the input shape while parameters are not. The default input shape is (1, 3, 340, 256) for 2D recognizer, (1, 3, 32, 340, 256) for 3D recognizer. (2) Some operators are not counted into FLOPs like GN and custom operators. Refer to `mmcv.cnn.get_model_complexity_info()` for details.

43.3 Model Conversion

43.3.1 MMPose model to ONNX (experimental)

`/tools/deployment/pytorch2onnx.py` is a script to convert model to [ONNX](#) format. It also supports comparing the output results between Pytorch and ONNX model for verification. Run `pip install onnx onnxruntime` first to install the dependency.

```
python tools/deployment/pytorch2onnx.py $CONFIG_PATH $CHECKPOINT_PATH --shape $SHAPE --
  ↪ verify
```

43.3.2 Prepare a model for publishing

`tools/publish_model.py` helps users to prepare their model for publishing.

Before you upload a model to AWS, you may want to:

(1) convert model weights to CPU tensors. (2) delete the optimizer states. (3) compute the hash of the checkpoint file and append the hash id to the filename.

```
python tools/publish_model.py ${INPUT_FILENAME} ${OUTPUT_FILENAME}
```

E.g.,

```
python tools/publish_model.py work_dirs/hrnet_w32_coco_256x192/latest.pth hrnet_w32_coco_
↪256x192
```

The final output filename will be `hrnet_w32_coco_256x192-{hash id}_{time_stamp}.pth`.

43.4 Model Serving

MMPose supports model serving with [TorchServe](#). You can serve an MMPose model via following steps:

43.4.1 1. Install TorchServe

Please follow the official installation guide of TorchServe: <https://github.com/pytorch/serve#install-torchserve-and-torch-model-archiver>

43.4.2 2. Convert model from MMPose to TorchServe

```
python tools/deployment/mmpose2torchserve.py \
    ${CONFIG_FILE} ${CHECKPOINT_FILE} \
    --output-folder ${MODEL_STORE} \
    --model-name ${MODEL_NAME}
```

Note: `${MODEL_STORE}` needs to be an absolute path to a folder.

A model file `${MODEL_NAME}.mar` will be generated and placed in the `${MODEL_STORE}` folder.

43.4.3 3. Deploy model serving

We introduce following 2 approaches to deploying the model serving.

Use TorchServe API

```
torchserve --start \
  --model-store ${MODEL_STORE} \
  --models ${MODEL_PATH1} [${MODEL_NAME}=${MODEL_PATH2} ... ]
```

Example:

```
# serve one model
torchserve --start --model-store /models --models hrnet=hrnet.mar

# serve all models in model-store
torchserve --start --model-store /models --models all
```

After executing the `torchserve` command above, TorchServe runs on your host, listening for inference requests. Check the [official docs](#) for more information.

Use `mmpose-serve` docker image

Build `mmpose-serve` docker image:

```
docker build -t mmpose-serve:latest docker/serve/
```

Run `mmpose-serve`:

Check the official docs for [running TorchServe with docker](#).

In order to run in GPU, you need to install [nvidia-docker](#). You can omit the `--gpus` argument in order to run in CPU.

Example:

```
docker run --rm \
  --cpus 8 \
  --gpus device=0 \
  -p8080:8080 -p8081:8081 -p8082:8082 \
  --mount type=bind,source=${MODEL_STORE},target=/home/model-server/model-store \
  mmpose-serve:latest
```

Read the [docs](#) about the Inference (8080), Management (8081) and Metrics (8082) APIs

43.4.4 4. Test deployment

You can use `tools/deployment/test_torchserver.py` to test the model serving. It will compare and visualize the result of `torchserver` and `pytorch`.

```
python tools/deployment/test_torchserver.py ${IMAGE_PATH} ${CONFIG_PATH} ${CHECKPOINT_
  →PATH} ${MODEL_NAME} --out-dir ${OUT_DIR}
```

Example:

```
python tools/deployment/test_torchserver.py \
  ls tests/data/coco/0000000000785.jpg \
  configs/body/2d_kpt_sview_rgb_img/topdown_heatmap/coco/hrnet_w48_coco_256x192.py \
  https://download.openmmlab.com/mmpose/top_down/hrnet/hrnet_w48_coco_256x192-b9e0b3ab_
  →20200708.pth \
```

(continues on next page)

(continued from previous page)

```
hrnet \  
--out-dir vis_results
```

43.5 Miscellaneous

43.5.1 Print the entire config

`tools/analysis/print_config.py` prints the whole config verbatim, expanding all its imports.

```
python tools/print_config.py ${CONFIG} [-h] [--options ${OPTIONS} [OPTIONS...]]
```

CHANGELOG

44.1 v0.29.0 (14/10/2022)

Highlights

- Support [DEKR](#), CVPR'2021 ([#1693](#)) @JinluZhang1126, @jin-s13, @Ben-Louis
- Support [CID](#), CVPR'2022 ([#1604](#)) @kennethwdk, @jin-s13, @Ben-Louis

Improvements

- Improve 3D pose estimation demo with viewpoint & height controlling and enriched input formats ([#1481](#), [#1490](#)) @pallgeuer, @liqikai9
- Improve smoothing of 3D video pose estimation ([#1479](#)) @darcula1993
- Improve dependency management for smoother installation ([#1491](#)) @danielbraun89
- Improve documentation quality ([#1680](#), [#1525](#), [#1516](#), [#1513](#), [#1511](#), [#1507](#), [#1506](#), [#1525](#)) @Ben-Louis, @Yulv-git, @jin-s13

Bug Fixes

- Update dependency versions of pre-commit-hooks ([#1706](#), [#1678](#)) @Ben-Louis, @liqikai9
- Fix missing data transforms in RLE configs ([#1632](#)) @Ben-Louis
- Fix a bug in `flipplr_joints` that causes error when keypoint visibility has float values ([#1589](#)) @walsvid
- Fix hsigmoid default parameters ([#1575](#)) @Ben-Louis
- Fix a bug in UDP decoding ([#1565](#)) @liqikai9
- Fix a bug in pose tracking demo with non-COCO dataset ([#1504](#)) @daixinghome
- Fix typos and unused contents in configs ([#1499](#), [#1496](#)) @liqikai9
- Fix a bug in the mesh demo ([#1482](#)) @zengwang430521

44.2 v0.28.1 (28/07/2022)

This release is meant to fix the compatibility with the latest mmcv v1.6.1

44.3 v0.28.0 (06/07/2022)

Highlights

- Support [TCFormer](#) backbone, CVPR'2022 (#1447, #1452) @zengwang430521
- Add [RLE](#) models on COCO dataset (#1424) @Indigo6, @Ben-Louis, @ly015
- Update swin models with better performance (#1467) @jin-s13

New Features

- Support [TCFormer](#) backbone, CVPR'2022 (#1447, #1452) @zengwang430521
- Add [RLE](#) models on COCO dataset (#1424) @Indigo6, @Ben-Louis, @ly015
- Support layer decay optimizer constructor and learning rate decay optimizer constructor (#1423) @jin-s13

Improvements

- Improve documentation quality (#1416, #1421, #1423, #1426, #1458, #1463) @ly015, @liqikai9
- Support installation by [mim](#) (#1425) @liqikai9
- Support PAVI logger (#1434) @EvelynWang-0423
- Add progress bar for some demos (#1454) @liqikai9
- Webcam API supports quick device setting in terminal commands (#1466) @ly015
- Update swin models with better performance (#1467) @jin-s13

Bug Fixes

- Rename `custom_hooks_config` to `custom_hooks` in configs to align with the documentation (#1427) @ly015
- Fix deadlock issue in Webcam API (#1430) @ly015
- Fix smoother configs in video 3D demo (#1457) @ly015

44.4 v0.27.0 (07/06/2022)

Highlights

- Support hand gesture recognition
 - Try the [demo](#) for gesture recognition
 - Learn more about the algorithm, dataset and [experiment results](#)
- Major upgrade to the Webcam API
 - Tutorials ([EN|zh_CN](#))
 - [API Reference](#)
 - [Demo](#)

New Features

- Support gesture recognition algorithm [MTUT CVPR'2019](#) and dataset [NVGesture CVPR'2016](#) (#1380) @Ben-Louis

Improvements

- Upgrade Webcam API and related documents (#1393, #1404, #1413) @ly015
- Support exporting COCO inference result without the annotation file (#1368) @liqikai9
- Replace markdownlint with mdformat in CI to avoid the dependence on ruby #1382 @ly015
- Improve documentation quality (#1385, #1394, #1395, #1408) @chubei-oppen, @ly015, @liqikai9

Bug Fixes

- Fix xywh->xyxy bbox conversion in dataset sanity check (#1367) @jin-s13
- Fix a bug in two-stage 3D keypoint demo (#1373) @ly015
- Fix out-dated settings in PVT configs (#1376) @ly015
- Fix myst settings for document compiling (#1381) @ly015
- Fix a bug in bbox transform (#1384) @ly015
- Fix inaccurate description of `min_keypoints` in tracking apis (#1398) @pallgeuer
- Fix warning with `torch.meshgrid` (#1402) @pallgeuer
- Remove redundant transformer modules from `mmpose.datasets.backbones.utils` (#1405) @ly015

44.5 v0.26.0 (05/05/2022)

Highlights

- Support [RLE \(Residual Log-likelihood Estimation\)](#), ICCV'2021 (#1259) @Indigo6, @ly015
- Support [Swin Transformer](#), ICCV'2021 (#1300) @yumendecc, @ly015
- Support [PVT](#), ICCV'2021 and [PVTv2](#), CVMJ'2022 (#1343) @zengwang430521
- Speed up inference and reduce CPU usage by optimizing the pre-processing pipeline (#1320) @chenxinfeng4, @liqikai9

New Features

- Support [RLE \(Residual Log-likelihood Estimation\)](#), ICCV'2021 (#1259) @Indigo6, @ly015
- Support [Swin Transformer](#), ICCV'2021 (#1300) @yumendecc, @ly015
- Support [PVT](#), ICCV'2021 and [PVTv2](#), CVMJ'2022 (#1343) @zengwang430521
- Support [FPN](#), CVPR'2017 (#1300) @yumendecc, @ly015

Improvements

- Speed up inference and reduce CPU usage by optimizing the pre-processing pipeline (#1320) @chenxinfeng4, @liqikai9
- Video demo supports models that requires multi-frame inputs (#1300) @liqikai9, @jin-s13
- Update benchmark regression list (#1328) @ly015, @liqikai9
- Remove unnecessary warnings in `TopDownPoseTrack18VideoDataset` (#1335) @liqikai9
- Improve documentation quality (#1313, #1305) @Ben-Louis, @ly015

- Update deprecating settings in configs (#1317) @ly015

Bug Fixes

- Fix a bug in human skeleton grouping that may skip the matching process unexpectedly when `ignore_to_much` is True (#1341) @daixinghome
- Fix a GPG key error that leads to CI failure (#1354) @ly015
- Fix bugs in distributed training script (#1338, #1298) @ly015
- Fix an upstream bug in xtccotools that causes incorrect AP(M) results (#1308) @jin-s13, @ly015
- Fix indentation errors in the colab tutorial (#1298) @YuanZi1501040205
- Fix incompatible model weight initialization with other OpenMMLab codebases (#1329) @274869388
- Fix HRNet FP16 checkpoints download URL (#1309) @YinAoXiong
- Fix typos in `body3d_two_stage_video_demo.py` (#1295) @mucozcan

Breaking Changes

- Refactor bbox processing in datasets and pipelines (#1311) @ly015, @Ben-Louis The bbox format conversion (xywh to center-scale) and random translation are moved from the dataset to the pipeline. The comparison between new and old version is as below:

```
...
# Data sample only contains bbox
rec.append({
    'bbox': obj['clean_bbox'][:4],
    ...
})
```

```
...
# Convert bbox from xywh to center-scale
center, scale = self._xywh2cs(*obj['clean_bbox'][:4])
# Data sample contains center and scale
rec.append({
    'bbox': obj['clean_bbox'][:4],
    'center': center,
    'scale': scale,
    ...
})
```

```
...
train_pipeline = [
    dict(type='LoadImageFromFile'),
    # Convert bbox from xywh to center-scale
    dict(type='TopDownGetBboxCenterScale', padding=1.25),
    # Randomly shift bbox center
    dict(type='TopDownRandomShiftBboxCenter', shift_factor=0.16, prob=0.3),
    ...
]
```

```
...
train_pipeline = [
    dict(type='LoadImageFromFile'),
```

(continues on next page)

(continued from previous page)

]

44.6 v0.25.0 (02/04/2022)

Highlights

- Support Shelf and Campus datasets with pre-trained VoxelPose models, “3D Pictorial Structures for Multiple Human Pose Estimation”, CVPR’2014 (#1225) @liqikai9, @wusize
- Add Smoother module for temporal smoothing of the pose estimation with configurable filters (#1127) @ailingzengzzz, @ly015
- Support SmoothNet for pose smoothing, “SmoothNet: A Plug-and-Play Network for Refining Human Poses in Videos”, arXiv’2021 (#1279) @ailingzengzzz, @ly015
- Add multiview 3D pose estimation demo (#1270) @wusize

New Features

- Support Shelf and Campus datasets with pre-trained VoxelPose models, “3D Pictorial Structures for Multiple Human Pose Estimation”, CVPR’2014 (#1225) @liqikai9, @wusize
- Add Smoother module for temporal smoothing of the pose estimation with configurable filters (#1127) @ailingzengzzz, @ly015
- Support SmoothNet for pose smoothing, “SmoothNet: A Plug-and-Play Network for Refining Human Poses in Videos”, arXiv’2021 (#1279) @ailingzengzzz, @ly015
- Add multiview 3D pose estimation demo (#1270) @wusize
- Support multi-machine distributed training (#1248) @ly015

Improvements

- Update HRFormer configs and checkpoints with relative position bias (#1245) @zengwang430521
- Support using different random seed for each distributed node (#1257, #1229) @ly015
- Improve documentation quality (#1275, #1255, #1258, #1249, #1247, #1240, #1235) @ly015, @jin-s13, @YoniChechik

Bug Fixes

- Fix keypoint index in RHD dataset meta information (#1265) @liqikai9
- Fix pre-commit hook unexpected behavior on Windows (#1282) @liqikai9
- Remove python-dev installation in CI (#1276) @ly015
- Unify hyphens in argument names in tools and demos (#1271) @ly015
- Fix ambiguous channel size in `channel_shuffle` that may cause exporting failure (#1242) @PINTO0309
- Fix a bug in Webcam API that causes single-class detectors fail (#1239) @674106399
- Fix the issue that `custom_hook` can not be set in configs (#1236) @bladrome
- Fix incompatible MMCV version in DockerFile (#raykindle)
- Skip invisible joints in visualization (#1228) @womeier

44.7 v0.24.0 (07/03/2022)

Highlights

- Support HRFormer “HRFormer: High-Resolution Vision Transformer for Dense Predict”, NeurIPS’2021 (#1203) @zengwang430521
- Support Windows installation with pip (#1213) @jin-s13, @ly015
- Add WebcamAPI documents (#1187) @ly015

New Features

- Support HRFormer “HRFormer: High-Resolution Vision Transformer for Dense Predict”, NeurIPS’2021 (#1203) @zengwang430521
- Support Windows installation with pip (#1213) @jin-s13, @ly015
- Support CPU training with mmcv < v1.4.4 (#1161) @EasonQYS, @ly015
- Add “Valentine Magic” demo with WebcamAPI (#1189, #1191) @liqikai9

Improvements

- Refactor multi-view 3D pose estimation framework towards better modularization and expansibility (#1196) @wusize
- Add WebcamAPI documents and tutorials (#1187) @ly015
- Refactor dataset evaluation interface to align with other OpenMMLab codebases (#1209) @ly015
- Add deprecation message for deploy tools since MMDeploy has supported MMPose (#1207) @QwQ2000
- Improve documentation quality (#1206, #1161) @ly015
- Switch to OpenMMLab official pre-commit-hook for copyright check (#1214) @ly015

Bug Fixes

- Fix hard-coded data collating and scattering in inference (#1175) @ly015
- Fix model configs on JHMDB dataset (#1188) @jin-s13
- Fix area calculation in pose tracking inference (#1197) @pallgeuer
- Fix registry scope conflict of module wrapper (#1204) @ly015
- Update MMCV installation in CI and documents (#1205)
- Fix incorrect color channel order in visualization functions (#1212) @ly015

44.8 v0.23.0 (11/02/2022)

Highlights

- Add **MMPose Webcam API**: A simple yet powerful tools to develop interactive webcam applications with MM-Pose functions. (#1178, #1173, #1173, #1143, #1094, #1133, #1098, #1160) @ly015, @jin-s13, @liqikai9, @wusize, @luminxu, @zengwang430521 @mzr1996

New Features

- Add **MMPose Webcam API**: A simple yet powerful tools to develop interactive webcam applications with MM-Pose functions. (#1178, #1173, #1173, #1143, #1094, #1133, #1098, #1160) @ly015, @jin-s13, @liqikai9, @wusize, @luminxu, @zengwang430521 @mzr1996

- Support ConcatDataset (#1139) @Canwang-sjtu
- Support CPU training and testing (#1157) @ly015

Improvements

- Add multi-processing configurations to speed up distributed training and testing (#1146) @ly015
- Add default runtime config (#1145)
- Upgrade isort in pre-commit hook (#1179) @liqikai9
- Update README and documents (#1171, #1167, #1153, #1149, #1148, #1147, #1140) @jin-s13, @wusize, @TommyZihao, @ly015

Bug Fixes

- Fix undeterministic behavior in pre-commit hooks (#1136) @jin-s13
- Deprecate the support for “python setup.py test” (#1179) @ly015
- Fix incompatible settings with MMCV on Hsigmoid default parameters (#1132) @ly015
- Fix alumentation installation (#1184) @BIGWangYuDong

44.9 v0.22.0 (04/01/2022)

Highlights

- Support VoxelPose “VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment”, ECCV’2020 (#1050) @wusize
- Support Soft Wing loss “Structure-Coherent Deep Feature Learning for Robust Face Alignment”, TIP’2021 (#1077) @jin-s13
- Support Adaptive Wing loss “Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression”, ICCV’2019 (#1072) @jin-s13

New Features

- Support VoxelPose “VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment”, ECCV’2020 (#1050) @wusize
- Support Soft Wing loss “Structure-Coherent Deep Feature Learning for Robust Face Alignment”, TIP’2021 (#1077) @jin-s13
- Support Adaptive Wing loss “Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression”, ICCV’2019 (#1072) @jin-s13
- Add LiteHRNet-18 Checkpoints trained on COCO. (#1120) @jin-s13

Improvements

- Improve documentation quality (#1115, #1111, #1105, #1087, #1086, #1085, #1084, #1083, #1124, #1070, #1068) @jin-s13, @liqikai9, @ly015
- Support CircleCI (#1074) @ly015
- Skip unit tests in CI when only document files were changed (#1074, #1041) @QwQ2000, @ly015
- Support file_client_args in LoadImageFromFile (#1076) @jin-s13

Bug Fixes

- Fix a bug in Dark UDP postprocessing that causes error when the channel number is large. (#1079, #1116) @X00123, @jin-s13
- Fix hard-coded sigmas in bottom-up image demo (#1107, #1101) @chenxinfeng4, @liqikai9
- Fix unstable checks in unit tests (#1112) @ly015
- Do not destroy NULL windows if `args.show==False` in demo scripts (#1104) @bladrome

44.10 v0.21.0 (06/12/2021)

Highlights

- Support “Learning Temporal Pose Estimation from Sparsely-Labeled Videos”, NeurIPS’2019 (#932, #1006, #1036, #1060) @liqikai9
- Add ViPNAS-MobileNetV3 models (#1025) @luminxu, @jin-s13
- Add *inference speed benchmark* (#1028, #1034, #1044) @liqikai9

New Features

- Support “Learning Temporal Pose Estimation from Sparsely-Labeled Videos”, NeurIPS’2019 (#932, #1006, #1036) @liqikai9
- Add ViPNAS-MobileNetV3 models (#1025) @luminxu, @jin-s13
- Add light-weight top-down models for whole-body keypoint detection (#1009, #1020, #1055) @luminxu, @ly015
- Add HRNet checkpoints with various settings on PoseTrack18 (#1035) @liqikai9

Improvements

- Add *inference speed benchmark* (#1028, #1034, #1044) @liqikai9
- Update model metafile format (#1001) @ly015
- Support minus output feature index in mobilenet_v3 (#1005) @luminxu
- Improve documentation quality (#1018, #1026, #1027, #1031, #1038, #1046, #1056, #1057) @edybk, @luminxu, @ly015, @jin-s13
- Set default random seed in training initialization (#1030) @ly015
- Skip CI when only specific files changed (#1041, #1059) @QwQ2000, @ly015
- Automatically cancel uncompleted action runs when new commit arrives (#1053) @ly015

Bug Fixes

- Update pose tracking demo to be compatible with latest mmtracking (#1014) @jin-s13
- Fix symlink creation failure when installed in Windows environments (#1039) @QwQ2000
- Fix AP-10K dataset sigmas (#1040) @jin-s13

44.11 v0.20.0 (01/11/2021)

Highlights

- Add AP-10K dataset for animal pose estimation (#987) @Annbless, @AlexTheBad, @jin-s13, @ly015
- Support TorchServe (#979) @ly015

New Features

- Add AP-10K dataset for animal pose estimation (#987) @Annbless, @AlexTheBad, @jin-s13, @ly015
- Add HRNetv2 checkpoints on 300W and COFW datasets (#980) @jin-s13
- Support TorchServe (#979) @ly015

Bug Fixes

- Fix some deprecated or risky settings in configs (#963, #976, #992) @jin-s13, @wusize
- Fix issues of default arguments of training and testing scripts (#970, #985) @liqikai9, @wusize
- Fix heatmap and tag size mismatch in bottom-up with UDP (#994) @wusize
- Fix python3.9 installation in CI (#983) @ly015
- Fix model zoo document integrity issue (#990) @jin-s13

Improvements

- Support non-square input shape for bottom-up (#991) @wusize
- Add image and video resources for demo (#971) @liqikai9
- Use CUDA docker images to accelerate CI (#973) @ly015
- Add codespell hook and fix detected typos (#977) @ly015

44.12 v0.19.0 (08/10/2021)

Highlights

- Add models for Associative Embedding with Hourglass network backbone (#906, #955) @jin-s13, @luminxu
- Support COCO-Wholebody-Face and COCO-Wholebody-Hand datasets (#813) @jin-s13, @innerlee, @luminxu
- Upgrade dataset interface (#901, #924) @jin-s13, @innerlee, @ly015, @liqikai9
- New style of documentation (#945) @ly015

New Features

- Add models for Associative Embedding with Hourglass network backbone (#906, #955) @jin-s13, @luminxu
- Support COCO-Wholebody-Face and COCO-Wholebody-Hand datasets (#813) @jin-s13, @innerlee, @luminxu
- Add pseudo-labeling tool to generate COCO style keypoint annotations with given bounding boxes (#928) @soltkreig
- New style of documentation (#945) @ly015

Bug Fixes

- Fix segmentation parsing in Macaque dataset preprocessing (#948) @jin-s13

- Fix dependencies that may lead to CI failure in downstream projects (#936, #953) @RangiLyu, @ly015
- Fix keypoint order in Human3.6M dataset (#940) @ttxskk
- Fix unstable image loading for Interhand2.6M (#913) @zengwang430521

Improvements

- Upgrade dataset interface (#901, #924) @jin-s13, @innerlee, @ly015, @liqikai9
- Improve demo usability and stability (#908, #934) @ly015
- Standardize model metafile format (#941) @ly015
- Support `persistent_worker` and several other arguments in configs (#946) @jin-s13
- Use MMCV root model registry to enable cross-project module building (#935) @RangiLyu
- Improve the document quality (#916, #909, #942, #913, #956) @jin-s13, @ly015, @bit-scientist, @zengwang430521
- Improve pull request template (#952, #954) @ly015

Breaking Changes

- Upgrade dataset interface (#901) @jin-s13, @innerlee, @ly015

44.13 v0.18.0 (01/09/2021)

Bug Fixes

- Fix redundant model weight loading in pytorch-to-onnx conversion (#850) @ly015
- Fix a bug in `update_model_index.py` that may cause pre-commit hook failure (#866) @ly015
- Fix a bug in `interhand_3d_head` (#890) @zengwang430521
- Fix pose tracking demo failure caused by out-of-date configs (#891)

Improvements

- Add automatic benchmark regression tools (#849, #880, #885) @liqikai9, @ly015
- Add copyright information and checking hook (#872)
- Add PR template (#875) @ly015
- Add citation information (#876) @ly015
- Add python3.9 in CI (#877, #883) @ly015
- Improve the quality of the documents (#845, #845, #848, #867, #870, #873, #896) @jin-s13, @ly015, @zhiqwang

44.14 v0.17.0 (06/08/2021)

Highlights

1. Support “Lite-HRNet: A Lightweight High-Resolution Network” CVPR’2021 (#733,#800) @jin-s13
2. Add 3d body mesh demo (#771) @zengwang430521
3. Add Chinese documentation (#787, #798, #799, #802, #804, #805, #815, #816, #817, #819, #839) @ly015, @luminxu, @jin-s13, @liqikai9, @zengwang430521
4. Add Colab Tutorial (#834) @ly015

New Features

- Support “Lite-HRNet: A Lightweight High-Resolution Network” CVPR’2021 (#733,#800) @jin-s13
- Add 3d body mesh demo (#771) @zengwang430521
- Add Chinese documentation (#787, #798, #799, #802, #804, #805, #815, #816, #817, #819, #839) @ly015, @luminxu, @jin-s13, @liqikai9, @zengwang430521
- Add Colab Tutorial (#834) @ly015
- Support training for InterHand v1.0 dataset (#761) @zengwang430521

Bug Fixes

- Fix mpII pckh@0.1 index (#773) @jin-s13
- Fix multi-node distributed test (#818) @ly015
- Fix docstring and init_weights error of ShuffleNetV1 (#814) @Junjun2016
- Fix imshow_bbox error when input bboxes is empty (#796) @ly015
- Fix model zoo doc generation (#778) @ly015
- Fix typo (#767), (#780, #782) @ly015, @jin-s13

Breaking Changes

- Use MMCV EvalHook (#686) @ly015

Improvements

- Add pytest.ini and fix docstring (#812) @jin-s13
- Update MSELoss (#829) @Ezra-Yu
- Move process_mmdet_results into inference.py (#831) @ly015
- Update resource limit (#783) @jin-s13
- Use COCO 2D pose model in 3D demo examples (#785) @ly015
- Change model zoo titles in the doc from center-aligned to left-aligned (#792, #797) @ly015
- Support MIM (#706, #794) @ly015
- Update out-of-date configs (#827) @jin-s13
- Remove opencv-python-headless dependency by albumentations (#833) @ly015
- Update QQ QR code in README_CN.md (#832) @ly015

44.15 v0.16.0 (02/07/2021)

Highlights

1. Support “ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search” CVPR’2021 (#742,#755).
2. Support MPI-INF-3DHP dataset (#683,#746,#751).
3. Add webcam demo tool (#729)
4. Add 3d body and hand pose estimation demo (#704, #727).

New Features

- Support “ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search” CVPR’2021 (#742,#755)
- Support MPI-INF-3DHP dataset (#683,#746,#751)
- Support Webcam demo (#729)
- Support Interhand 3d demo (#704)
- Support 3d pose video demo (#727)
- Support H36m dataset for 2d pose estimation (#709, #735)
- Add scripts to generate mim metafile (#749)

Bug Fixes

- Fix typos (#692,#696,#697,#698,#712,#718,#728)
- Change model download links from http to https (#716)

Breaking Changes

- Switch to MMCV MODEL_REGISTRY (#669)

Improvements

- Refactor MeshMixDataset (#752)
- Rename ‘GaussianHeatMap’ to ‘GaussianHeatmap’ (#745)
- Update out-of-date configs (#734)
- Improve compatibility for breaking changes (#731)
- Enable to control radius and thickness in visualization (#722)
- Add regex dependency (#720)

44.16 v0.15.0 (02/06/2021)

Highlights

1. Support 3d video pose estimation (VideoPose3D).
2. Support 3d hand pose estimation (InterNet).
3. Improve presentation of modelzoo.

New Features

- Support “InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image” (ECCV’20) (#624)

- Support “3D human pose estimation in video with temporal convolutions and semi-supervised training” (CVPR’19) (#602, #681)
- Support 3d pose estimation demo (#653, #670)
- Support bottom-up whole-body pose estimation (#689)
- Support mmcli (#634)

Bug Fixes

- Fix opencv compatibility (#635)
- Fix demo with UDP (#637)
- Fix bottom-up model onnx conversion (#680)
- Fix GPU_IDS in distributed training (#668)
- Fix MANIFEST.in (#641, #657)
- Fix docs (#643, #684, #688, #690, #692)

Breaking Changes

- Reorganize configs by tasks, algorithms, datasets, and techniques (#647)
- Rename heads and detectors (#667)

Improvements

- Add radius and thickness parameters in visualization (#638)
- Add trans_prob parameter in TopDownRandomTranslation (#650)
- Switch to MMCV MODEL_REGISTRY (#669)
- Update dependencies (#674, #676)

44.17 v0.14.0 (06/05/2021)

Highlights

1. Support animal pose estimation with 7 popular datasets.
2. Support “A simple yet effective baseline for 3d human pose estimation” (ICCV’17).

New Features

- Support “A simple yet effective baseline for 3d human pose estimation” (ICCV’17) (#554, #558, #566, #570, #589)
- Support animal pose estimation (#559, #561, #563, #571, #603, #605)
- Support Horse-10 dataset (#561), MacaquePose dataset (#561), Vinegar Fly dataset (#561), Desert Locust dataset (#561), Grevy’s Zebra dataset (#561), ATRW dataset (#571), and Animal-Pose dataset (#603)
- Support bottom-up pose tracking demo (#574)
- Support FP16 training (#584, #616, #626)
- Support NMS for bottom-up (#609)

Bug Fixes

- Fix bugs in the top-down demo, when there are no people in the images (#569).
- Fix the links in the doc (#612)

Improvements

- Speed up top-down inference (#560)
- Update github CI (#562, #564)
- Update Readme (#578, #579, #580, #592, #599, #600, #607)
- Update Faq (#587, #610)

44.18 v0.13.0 (31/03/2021)

Highlights

1. Support Wingloss.
2. Support RHD hand dataset.

New Features

- Support Wingloss (#482)
- Support RHD hand dataset (#523, #551)
- Support Human3.6m dataset for 3d keypoint detection (#518, #527)
- Support TCN model for 3d keypoint detection (#521, #522)
- Support Interhand3D model for 3d hand detection (#536)
- Support Multi-task detector (#480)

Bug Fixes

- Fix PCKh@0.1 calculation (#516)
- Fix unittest (#529)
- Fix circular importing (#542)
- Fix bugs in bottom-up keypoint score (#548)

Improvements

- Update config & checkpoints (#525, #546)
- Fix typos (#514, #519, #532, #537,)
- Speed up post processing (#535)
- Update mmcv version dependency (#544)

44.19 v0.12.0 (28/02/2021)

Highlights

1. Support DeepPose algorithm.

New Features

- Support DeepPose algorithm (#446, #461)
- Support interhand3d dataset (#468)
- Support Albumentation pipeline (#469)

- Support PhotometricDistortion pipeline (#485)
- Set seed option for training (#493)
- Add demos for face keypoint detection (#502)

Bug Fixes

- Change channel order according to configs (#504)
- Fix num_factors in UDP encoding (#495)
- Fix configs (#456)

Breaking Changes

- Refactor configs for wholebody pose estimation (#487, #491)
- Rename decode function for heads (#481)

Improvements

- Update config & checkpoints (#453, #484, #487)
- Add README in Chinese (#462)
- Add tutorials about configs (#465)
- Add demo videos for various tasks (#499, #503)
- Update docs about MMPose installation (#467, #505)
- Rename stat.py to stats.py (#483)
- Fix typos (#463, #464, #477, #481)
- latex to bibtex (#471)
- Update FAQ (#466)

44.20 v0.11.0 (31/01/2021)

Highlights

1. Support fashion landmark detection.
2. Support face keypoint detection.
3. Support pose tracking with MMTracking.

New Features

- Support fashion landmark detection (DeepFashion) (#413)
- Support face keypoint detection (300W, AFLW, COFW, WFLW) (#367)
- Support pose tracking demo with MMTracking (#427)
- Support face demo (#443)
- Support AIC dataset for bottom-up methods (#438, #449)

Bug Fixes

- Fix multi-batch training (#434)
- Fix sigmas in AIC dataset (#441)

- Fix config file (#420)

Breaking Changes

- Refactor Heads (#382)

Improvements

- Update readme (#409, #412, #415, #416, #419, #421, #422, #424, #425, #435, #436, #437, #444, #445)
- Add GAP (global average pooling) neck (#414)
- Speed up (#411, #423)
- Support COCO test-dev test (#433)

44.21 v0.10.0 (31/12/2020)

Highlights

1. Support more human pose estimation methods.
 - UDP
2. Support pose tracking.
3. Support multi-batch inference.
4. Add some useful tools, including `analyze_logs`, `get_flops`, `print_config`.
5. Support more backbone networks.
 - ResNest
 - VGG

New Features

- Support UDP (#353, #371, #402)
- Support multi-batch inference (#390)
- Support MHP dataset (#386)
- Support pose tracking demo (#380)
- Support mpII-trb demo (#372)
- Support mobilenet for hand pose estimation (#377)
- Support ResNest backbone (#370)
- Support VGG backbone (#370)
- Add some useful tools, including `analyze_logs`, `get_flops`, `print_config` (#324)

Bug Fixes

- Fix bugs in pck evaluation (#328)
- Fix model download links in README (#396, #397)
- Fix CrowdPose annotations and update benchmarks (#384)
- Fix modelzoo stat (#354, #360, #362)
- Fix config files for aic datasets (#340)

Breaking Changes

- Rename `image_thr` to `det_bbox_thr` for top-down methods.

Improvements

- Organize the readme files (#398, #399, #400)
- Check linting for markdown (#379)
- Add `faq.md` (#350)
- Remove PyTorch 1.4 in CI (#338)
- Add pypi badge in readme (#329)

44.22 v0.9.0 (30/11/2020)**Highlights**

1. Support more human pose estimation methods.
 - MSPN
 - RSN
2. Support video pose estimation datasets.
 - sub-JHMDB
3. Support Onnx model conversion.

New Features

- Support MSPN (#278)
- Support RSN (#221, #318)
- Support new post-processing method for MSPN & RSN (#288)
- Support sub-JHMDB dataset (#292)
- Support urls for pre-trained models in config files (#232)
- Support Onnx (#305)

Bug Fixes

- Fix model download links in README (#255, #315)

Breaking Changes

- `post_process=True|False` and `unbiased_decoding=True|False` are deprecated, use `post_process=None|default|unbiased` etc. instead (#288)

Improvements

- Enrich the model zoo (#256, #320)
- Set the default `map_location` as 'cpu' to reduce gpu memory cost (#227)
- Support return heatmaps and backbone features for bottom-up models (#229)
- Upgrade mmcv maximum & minimum version (#269, #313)
- Automatically add modelzoo statistics to readthedocs (#252)

- Fix Pylint issues (#258, #259, #260, #262, #265, #267, #268, #270, #271, #272, #273, #275, #276, #283, #285, #293, #294, #295)
- Improve README (#226, #257, #264, #280, #296)
- Support PyTorch 1.7 in CI (#274)
- Add docs/tutorials for running demos (#263)

44.23 v0.8.0 (31/10/2020)

Highlights

1. Support more human pose estimation datasets.
 - [CrowdPose](#)
 - [PoseTrack18](#)
2. Support more 2D hand keypoint estimation datasets.
 - [InterHand2.6](#)
3. Support adversarial training for 3D human shape recovery.
4. Support multi-stage losses.
5. Support mpII demo.

New Features

- Support [CrowdPose](#) dataset (#195)
- Support [PoseTrack18](#) dataset (#220)
- Support [InterHand2.6](#) dataset (#202)
- Support adversarial training for 3D human shape recovery (#192)
- Support multi-stage losses (#204)

Bug Fixes

- Fix config files (#190)

Improvements

- Add mpII demo (#216)
- Improve README (#181, #183, #208)
- Support return heatmaps and backbone features (#196, #212)
- Support different return formats of mmdetection models (#217)

44.24 v0.7.0 (30/9/2020)

Highlights

1. Support HMR for 3D human shape recovery.
2. Support WholeBody human pose estimation.
 - [COCO-WholeBody](#)
3. Support more 2D hand keypoint estimation datasets.
 - [Frei-hand](#)
 - [CMU Panoptic HandDB](#)
4. Add more popular backbones & enrich the [modelzoo](#)
 - [ShuffleNetv2](#)
5. Support hand demo and whole-body demo.

New Features

- Support HMR for 3D human shape recovery ([#157](#), [#160](#), [#161](#), [#162](#))
- Support [COCO-WholeBody](#) dataset ([#133](#))
- Support [Frei-hand](#) dataset ([#125](#))
- Support [CMU Panoptic HandDB](#) dataset ([#144](#))
- Support H36M dataset ([#159](#))
- Support [ShuffleNetv2](#) ([#139](#))
- Support saving best models based on key indicator ([#127](#))

Bug Fixes

- Fix typos in docs ([#121](#))
- Fix assertion ([#142](#))

Improvements

- Add tools to transform .mat format to .json format ([#126](#))
- Add hand demo ([#115](#))
- Add whole-body demo ([#163](#))
- Reuse mmcv utility function and update version files ([#135](#), [#137](#))
- Enrich the [modelzoo](#) ([#147](#), [#169](#))
- Improve docs ([#174](#), [#175](#), [#178](#))
- Improve README ([#176](#))
- Improve version.py ([#173](#))

44.25 v0.6.0 (31/8/2020)

Highlights

1. Add more popular backbones & enrich the [modelzoo](#)
 - ResNext
 - SEResNet
 - ResNetV1D
 - MobileNetv2
 - ShuffleNetv1
 - CPM (Convolutional Pose Machine)
2. Add more popular datasets:
 - [AIChallenger](#)
 - [MPII](#)
 - [MPII-TRB](#)
 - [OCHuman](#)
3. Support 2d hand keypoint estimation.
 - [OneHand10K](#)
4. Support bottom-up inference.

New Features

- Support [OneHand10K](#) dataset (#52)
- Support [MPII](#) dataset (#55)
- Support [MPII-TRB](#) dataset (#19, #47, #48)
- Support [OCHuman](#) dataset (#70)
- Support [AIChallenger](#) dataset (#87)
- Support multiple backbones (#26)
- Support CPM model (#56)

Bug Fixes

- Fix configs for MPII & MPII-TRB datasets (#93)
- Fix the bug of missing `test_pipeline` in configs (#14)
- Fix typos (#27, #28, #50, #53, #63)

Improvements

- Update benchmark (#93)
- Add Dockerfile (#44)
- Improve unittest coverage and minor fix (#18)
- Support CPUs for train/val/demo (#34)
- Support bottom-up demo (#69)

- Add tools to publish model (#62)
- Enrich the modelzoo (#64, #68, #82)

44.26 v0.5.0 (21/7/2020)

Highlights

- MMPose is released.

Main Features

- Support both top-down and bottom-up pose estimation approaches.
- Achieve higher training efficiency and higher accuracy than other popular codebases (e.g. AlphaPose, HRNet)
- Support various backbone models: ResNet, HRNet, SCNet, Houghlass and HigherHRNet.
- *FAQ*
 - *Installation*
 - *Coding*
 - *Data*
 - *Training*
 - *Evaluation*
 - *Inference*
 - *Deployment*

FAQ

We list some common issues faced by many users and their corresponding solutions here. Feel free to enrich the list if you find any frequent issues and have ways to help others to solve them. If the contents here do not cover your issue, please create an issue using the [provided templates](#) and make sure you fill in all required information in the template.

45.1 Installation

- Compatibility issue between MMCV and MMDetection; “AssertionError: MMCV==xxx is used but incompatible. Please install mmcv>=xxx, <=xxx.”

Compatible MMDetection and MMCV versions are shown as below. Please choose the correct version of MMCV to avoid installation issues.

- **Unable to install xtcocotools**

1. Try to install it using pypi manually `pip install xtcocotools`.
2. If step1 does not work. Try to install it from [source](#).

```
git clone https://github.com/jin-s13/xtcocoapi
cd xtcocoapi
python setup.py install
```

- **No matching distribution found for xtcocotools>=1.6**

1. Install cython by `pip install cython`.
2. Install xtcocotools from [source](#).

```
git clone https://github.com/jin-s13/xtcocoapi
cd xtcocoapi
python setup.py install
```

- “No module named ‘mmcv.ops’”; “No module named ‘mmcv._ext’”

1. Uninstall existing mmcv in the environment using `pip uninstall mmcv`.
2. Install mmcv-full following the [installation instruction](#).

45.2 Coding

- Do I need to reinstall mmpose after some code modifications?

If you follow *the best practice* and install mmpose from source, any local modifications made to the code will take effect without reinstallation.

- How to develop with multiple MMPose versions?

Generally speaking, we recommend to use different virtual environments to manage MMPose in different working directories. However, you can also use the same environment to develop MMPose in different folders, like `mmpose-0.26.0`, `mmpose-0.25.0`. When you run the train or test shell script, it will adopt the mmpose package in the current folder. And when you run other Python script, you can also add `PYTHONPATH=`pwd`` at the beginning of your command to use the package in the current folder.

Conversely, to use the default MMPose installed in the environment rather than the one you are working with, you can remove the following line in those shell scripts:

```
PYTHONPATH="$(dirname $0)/..":$PYTHONPATH
```

45.3 Data

- **How to convert my 2d keypoint dataset to coco-type?**

You may refer to this conversion [tool](#) to prepare your data. Here is an [example](#) of the coco-type json. In the coco-type json, we need “categories”, “annotations” and “images”. “categories” contain some basic information of the dataset, e.g. class name and keypoint names. “images” contain image-level information. We need “id”, “file_name”, “height”, “width”. Others are optional. Note: (1) It is okay that “id”s are not continuous or not sorted (e.g. 1000, 40, 352, 333 ...).

“annotations” contain instance-level information. We need “image_id”, “id”, “keypoints”, “num_keypoints”, “bbox”, “iscrowd”, “area”, “category_id”. Others are optional. Note: (1) “num_keypoints” means the number of visible keypoints. (2) By default, please set “iscrowd: 0”. (3) “area” can be calculated using the bbox ($\text{area} = w * h$) (4) Simply set “category_id: 1”. (5) The “image_id” in “annotations” should match the “id” in “images”.

- **What if my custom dataset does not have bounding box label?**

We can estimate the bounding box of a person as the minimal box that tightly bounds all the keypoints.

- **What if my custom dataset does not have segmentation label?**

Just set the area of the person as the area of the bounding boxes. During evaluation, please set `use_area=False` as in this [example](#).

- **What is COCO_val2017_detections_AP_H_56_person.json? Can I train pose models without it?**

“COCO_val2017_detections_AP_H_56_person.json” contains the “detected” human bounding boxes for COCO validation set, which are generated by FasterRCNN. One can choose to use gt bounding boxes to evaluate models, by setting `use_gt_bbox=True` and `bbox_file=''`. Or one can use detected boxes to evaluate the generalizability of models, by setting `use_gt_bbox=False` and `bbox_file='COCO_val2017_detections_AP_H_56_person.json'`.

45.4 Training

- **RuntimeError: Address already in use**

Set the environment variables MASTER_PORT=XXX. For example, MASTER_PORT=29517 GPUS=16 GPUS_PER_NODE=8 CPUS_PER_TASK=2 ./tools/slurm_train.sh Test res50 configs/body/2D_Kpt_SV_RGB_Img/topdown_hm/coco/res50_coco_256x192.py work_dirs/res50_coco_256x192

- **“Unexpected keys in source state dict” when loading pre-trained weights**

It’s normal that some layers in the pretrained model are not used in the pose model. ImageNet-pretrained classification network and the pose network may have different architectures (e.g. no classification head). So some unexpected keys in source state dict is actually expected.

- **How to use trained models for backbone pre-training ?**

Refer to [Use Pre-Trained Model](#), in order to use the pre-trained model for the whole network (backbone + head), the new config adds the link of pre-trained models in the load_from.

And to use backbone for pre-training, you can change pretrained value in the backbone dict of config files to the checkpoint path / url. When training, the unexpected keys will be ignored.

- **How to visualize the training accuracy/loss curves in real-time ?**

Use TensorboardLoggerHook in log_config like

```
log_config=dict(interval=20, hooks=[dict(type='TensorboardLoggerHook')])
```

You can refer to [customize_runtime.md](#) and the example [config](#).

- **Log info is NOT printed**

Use smaller log interval. For example, change interval=50 to interval=1 in the [config](#).

- **How to fix stages of backbone when finetuning a model ?**

You can refer to [def _freeze_stages\(\)](#) and [frozen_stages](#), reminding to set find_unused_parameters = True in config files for distributed training or testing.

45.5 Evaluation

- **How to evaluate on MPII test dataset?**

Since we do not have the ground-truth for test dataset, we cannot evaluate it ‘locally’. If you would like to evaluate the performance on test set, you have to upload the pred.mat (which is generated during testing) to the official server via email, according to [the MPII guideline](#).

- **For top-down 2d pose estimation, why predicted joint coordinates can be out of the bounding box (bbox)?**

We do not directly use the bbox to crop the image. Bbox will be first transformed to center & scale, and the scale will be multiplied by a factor (1.25) to include some context. If the ratio of width/height is different from that of model input (possibly 192/256), we will adjust the bbox. You can refer to [the code](#) for more details.

45.6 Inference

- **How to run mmpose on CPU?**

Run demos with `--device=cpu`.

- **How to speed up inference?**

For top-down models, try to edit the config file. For example,

1. set `flip_test=False` in `topdown-res50`.
2. set `post_process='default'` in `topdown-res50`.
3. use faster human bounding box detector, see [MMDetection](#).

For bottom-up models, try to edit the config file. For example,

1. set `flip_test=False` in `AE-res50`.
2. set `adjust=False` in `AE-res50`.
3. set `refine=False` in `AE-res50`.
4. use smaller input image size in `AE-res50`.

45.7 Deployment

- **Why is the onnx model converted by mmpose throwing error when converting to other frameworks such as TensorRT?**

For now, we can only make sure that models in mmpose are onnx-compatible. However, some operations in onnx may be unsupported by your target framework for deployment, e.g. TensorRT in [this issue](#).

And please note that `pytorch2onnx` in MMPose is no longer maintained and will be deprecated in the future. We have [MMDeploy](#) to support model deployment for all OpenMMLab codebases including MMPose. You can find details about supported models and user guides in their [documentation](#), and raise issues to request support for the models you would like to use.

MMPOSE.APIS

`mmpose.apis.collect_multi_frames(video, frame_id, indices, online=False)`

Collect multi frames from the video.

Parameters

- **video** (*mmcv.VideoReader*) – A VideoReader of the input video file.
- **frame_id** (*int*) – index of the current frame
- **indices** (*list(int)*) – index offsets of the frames to collect
- **online** (*bool*) – inference mode, if set to True, can not use future frame information.

Returns multi frames collected from the input video file.

Return type *list(ndarray)*

`mmpose.apis.extract_pose_sequence(pose_results, frame_idx, causal, seq_len, step=1)`

Extract the target frame from 2D pose results, and pad the sequence to a fixed length.

Parameters

- **pose_results** (*list[list[dict]]*) – Multi-frame pose detection results stored in a nested list. Each element of the outer list is the pose detection results of a single frame, and each element of the inner list is the pose information of one person, which contains:
 - **keypoints** (*ndarray[K, 2 or 3]*): x, y, [score]
 - **track_id** (*int*): unique id of each person, required when `with_track_id==True`.
 - **bbox** ((4,) or (5,)): left, right, top, bottom, [score]
- **frame_idx** (*int*) – The index of the frame in the original video.
- **causal** (*bool*) – If True, the target frame is the last frame in a sequence. Otherwise, the target frame is in the middle of a sequence.
- **seq_len** (*int*) – The number of frames in the input sequence.
- **step** (*int*) – Step size to extract frames from the video.

Returns Multi-frame pose detection results stored in a nested list with a length of `seq_len`.

Return type *list[list[dict]]*

`mmpose.apis.get_track_id(results, results_last, next_id, min_keypoints=3, use oks=False, tracking_thr=0.3, use_one_euro=False, fps=None, sigmas=None)`

Get track id for each person instance on the current frame.

Parameters

- **results** (*list[dict]*) – The bbox & pose results of the current frame (bbox_result, pose_result).
- **results_last** (*list[dict]*, *optional*) – The bbox & pose & track_id info of the last frame (bbox_result, pose_result, track_id). None is equivalent to an empty result list. Default: None
- **next_id** (*int*) – The track id for the new person instance.
- **min_keypoints** (*int*) – Minimum number of keypoints recognized as person. 0 means no minimum threshold required. Default: 3.
- **use_oks** (*bool*) – Flag to using oks tracking. default: False.
- **tracking_thr** (*float*) – The threshold for tracking.
- **use_one_euro** (*bool*) – Option to use one-euro-filter. default: False.
- **fps** (*optional*) – Parameters that d_cutoff when one-euro-filter is used as a video input
- **sigmas** (*np.ndarray*) – Standard deviation of keypoint labelling. It is necessary for oks_iou tracking (*use_oks==True*). It will be use sigmas of COCO as default if it is set to None. Default is None.

Returns

- results (list[dict]): The bbox & pose & track_id info of the current frame (bbox_result, pose_result, track_id).
- next_id (int): The track id for the new person instance.

Return type tuple

`mmpose.apis.inference_bottom_up_pose_model(model, img_or_path, dataset='BottomUpCocoDataset', dataset_info=None, pose_nms_thr=0.9, return_heatmap=False, outputs=None)`

Inference a single image with a bottom-up pose model.

Note:

- num_people: P
 - num_keypoints: K
 - bbox height: H
 - bbox width: W
-

Parameters

- **model** (*nn.Module*) – The loaded pose model.
- **img_or_path** (*str* / *np.ndarray*) – Image filename or loaded image.
- **dataset** (*str*) – Dataset name, e.g. 'BottomUpCocoDataset'. It is deprecated. Please use dataset_info instead.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **pose_nms_thr** (*float*) – retain oks overlap < pose_nms_thr, default: 0.9.
- **return_heatmap** (*bool*) – Flag to return heatmap, default: False.

- **outputs** (*list(str) | tuple(str)*) – Names of layers whose outputs need to be returned, default: None.

Returns

- **pose_results** (*list[np.ndarray]*): The predicted pose info. The length of the list is the number of people (P). Each item in the list is a ndarray, containing each person's pose (*np.ndarray[Kx3]*): x, y, score.
- **returned_outputs** (*list[dict[np.ndarray[N, K, H, W] | torch.Tensor[N, K, H, W]]]*): Output feature maps from layers specified in *outputs*. Includes 'heatmap' if *return_heatmap* is True.

Return type tuple

`mmpose.apis.inference_interhand_3d_model(model, img_or_path, det_results, bbox_thr=None, format='xywh', dataset='InterHand3DDataset')`

Inference a single image with a list of hand bounding boxes.

Note:

- num_bboxes: N
 - num_keypoints: K
-

Parameters

- **model** (*nn.Module*) – The loaded pose model.
- **img_or_path** (*str | np.ndarray*) – Image filename or loaded image.
- **det_results** (*list[dict]*) – The 2D bbox sequences stored in a list. Each element of the list is the bbox of one person, whose shape is (*ndarray[4 or 5]*), containing 4 box coordinates (and score).
- **dataset** (*str*) – Dataset name.
- **format** – bbox format ('xyxy' | 'xywh'). Default: 'xywh'. 'xyxy' means (left, top, right, bottom), 'xywh' means (left, top, width, height).

Returns 3D pose inference results. Each element is the result of an instance, which contains the predicted 3D keypoints with shape (*ndarray[K,3]*). If there is no valid instance, an empty list will be returned.

Return type list[dict]

`mmpose.apis.inference_mesh_model(model, img_or_path, det_results, bbox_thr=None, format='xywh', dataset='MeshH36MDataset')`

Inference a single image with a list of bounding boxes.

Note:

- num_bboxes: N
 - num_keypoints: K
 - num_vertices: V
 - num_faces: F
-

Parameters

- **model** (*nn.Module*) – The loaded pose model.
- **img_or_path** (*str* | *np.ndarray*) – Image filename or loaded image.
- **det_results** (*list[dict]*) – The 2D bbox sequences stored in a list. Each element of the list is the bbox of one person. “bbox” (*ndarray*[4 or 5]): The person bounding box, which contains 4 box coordinates (and score).
- **bbox_thr** (*float* | *None*) – Threshold for bounding boxes. Only bboxes with higher scores will be fed into the pose detector. If *bbox_thr* is *None*, all boxes will be used.
- **format** (*str*) – bbox format (‘xyxy’ | ‘xywh’). Default: ‘xywh’.
 - ‘xyxy’ means (left, top, right, bottom),
 - ‘xywh’ means (left, top, width, height).
- **dataset** (*str*) – Dataset name.

Returns

3D pose inference results. Each element is the result of an instance, which contains:

- ‘bbox’ (*ndarray*[4]): instance bounding bbox
- ‘center’ (*ndarray*[2]): bbox center
- ‘scale’ (*ndarray*[2]): bbox scale
- ‘keypoints_3d’ (*ndarray*[K,3]): predicted 3D keypoints
- ‘camera’ (*ndarray*[3]): camera parameters
- ‘vertices’ (*ndarray*[V, 3]): predicted 3D vertices
- ‘faces’ (*ndarray*[F, 3]): mesh faces

If there is no valid instance, an empty list will be returned.

Return type *list[dict]*

`mmpose.apis.inference_pose_lifter_model(model, pose_results_2d, dataset=None, dataset_info=None, with_track_id=True, image_size=None, norm_pose_2d=False)`

Inference 3D pose from 2D pose sequences using a pose lifter model.

Parameters

- **model** (*nn.Module*) – The loaded pose lifter model
- **pose_results_2d** (*list[list[dict]]*) – The 2D pose sequences stored in a nested list. Each element of the outer list is the 2D pose results of a single frame, and each element of the inner list is the 2D pose of one person, which contains:
 - “keypoints” (*ndarray*[K, 2 or 3]): x, y, [score]
 - “track_id” (int)
- **dataset** (*str*) – Dataset name, e.g. ‘Body3DH36MDataset’
- **with_track_id** – If *True*, the element in *pose_results_2d* is expected to contain “track_id”, which will be used to gather the pose sequence of a person from multiple frames. Otherwise, the pose results in each frame are expected to have a consistent number and order of identities. Default is *True*.

- **image_size** (*tuple/list*) – image width, image height. If None, image size will not be contained in dict data.
- **norm_pose_2d** (*bool*) – If True, scale the bbox (along with the 2D pose) to the average bbox scale of the dataset, and move the bbox (along with the 2D pose) to the average bbox center of the dataset.

Returns

3D pose inference results. Each element is the result of an instance, which contains:

- "keypoints_3d" (*ndarray[K, 3]*): predicted 3D keypoints
- "keypoints" (*ndarray[K, 2 or 3]*): from the last frame in `pose_results_2d`.
- "track_id" (*int*): from the last frame in `pose_results_2d`. If there is no valid instance, an empty list will be returned.

Return type `list[dict]`

```
mmpose.apis.inference_top_down_pose_model(model, imgs_or_paths, person_results=None,
                                           bbox_thr=None, format='xywh',
                                           dataset='TopDownCocoDataset', dataset_info=None,
                                           return_heatmap=False, outputs=None)
```

Inference a single image with a list of person bounding boxes. Support single-frame and multi-frame inference setting.

Note:

- num_frames: F
 - num_people: P
 - num_keypoints: K
 - bbox height: H
 - bbox width: W
-

Parameters

- **model** (*nn.Module*) – The loaded pose model.
- **imgs_or_paths** (*str | np.ndarray | list(str) | list(np.ndarray)*) – Image filename(s) or loaded image(s).
- **person_results** (*list(dict), optional*) – a list of detected persons that contains bbox and/or track_id:
 - **bbox (4,) or (5,)**: The person bounding box, which contains 4 box coordinates (and score).
 - **track_id (int)**: The unique id for each human instance. If not provided, a dummy person result with a bbox covering the entire image will be used. Default: None.
- **bbox_thr** (*float | None*) – Threshold for bounding boxes. Only bboxes with higher scores will be fed into the pose detector. If `bbox_thr` is None, all boxes will be used.
- **format** (*str*) – bbox format ('xyxy' | 'xywh'). Default: 'xywh'.
 - *xyxy* means (left, top, right, bottom),
 - *xywh* means (left, top, width, height).

- **dataset** (*str*) – Dataset name, e.g. ‘TopDownCocoDataset’. It is deprecated. Please use `dataset_info` instead.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **return_heatmap** (*bool*) – Flag to return heatmap, default: False
- **outputs** (*list(str) | tuple(str)*) – Names of layers whose outputs need to be returned. Default: None.

Returns

- **pose_results** (*list[dict]*): The bbox & pose info. Each item in the list is a dictionary, containing the bbox: (left, top, right, bottom, [score]) and the pose (*ndarray*[Kx3]): x, y, score.
- **returned_outputs** (*list[dict[np.ndarray[N, K, H, W] | torch.Tensor[N, K, H, W]]]*): Output feature maps from layers specified in *outputs*. Includes ‘heatmap’ if *return_heatmap* is True.

Return type `tuple`**mmpose.apis.init_pose_model**(*config, checkpoint=None, device='cuda:0'*)

Initialize a pose model from config file.

Parameters

- **config** (*str* or *mmcv.Config*) – Config file path or the config object.
- **checkpoint** (*str, optional*) – Checkpoint path. If left as None, the model will not load any weights.

Returns The constructed detector.**Return type** `nn.Module`**mmpose.apis.init_random_seed**(*seed=None, device='cuda'*)

Initialize random seed.

If the seed is not set, the seed will be automatically randomized, and then broadcast to all processes to prevent some potential bugs.

Parameters

- **seed** (*int, Optional*) – The seed. Default to None.
- **device** (*str*) – The device where the seed will be put on. Default to ‘cuda’.

Returns Seed to be used.**Return type** `int`**mmpose.apis.multi_gpu_test**(*model, data_loader, tmpdir=None, gpu_collect=False*)

Test model with multiple gpus.

This method tests model with multiple gpus and collects the results under two different modes: gpu and cpu modes. By setting ‘gpu_collect=True’ it encodes results to gpu tensors and use gpu communication for results collection. On cpu mode it saves the results on different gpus to ‘tmpdir’ and collects them by the rank 0 worker.

Parameters

- **model** (*nn.Module*) – Model to be tested.
- **data_loader** (*nn.DataLoader*) – Pytorch data loader.
- **tmpdir** (*str*) – Path of directory to save the temporary results from different gpus under cpu mode.
- **gpu_collect** (*bool*) – Option to use either gpu or cpu to collect results.

Returns The prediction results.

Return type list

`mmpose.apis.process_mmdet_results(mmdet_results, cat_id=1)`

Process mmdet results, and return a list of bboxes.

Parameters

- **mmdet_results** (*list/tuple*) – mmdet results.
- **cat_id** (*int*) – category id (default: 1 for human)

Returns a list of detected bounding boxes

Return type person_results (list)

`mmpose.apis.single_gpu_test(model, data_loader)`

Test model with a single gpu.

This method tests model with a single gpu and displays test progress bar.

Parameters

- **model** (*nn.Module*) – Model to be tested.
- **data_loader** (*nn.DataLoader*) – Pytorch data loader.

Returns The prediction results.

Return type list

`mmpose.apis.train_model(model, dataset, cfg, distributed=False, validate=False, timestamp=None, meta=None)`

Train model entry function.

Parameters

- **model** (*nn.Module*) – The model to be trained.
- **dataset** (*Dataset*) – Train dataset.
- **cfg** (*dict*) – The config dict for training.
- **distributed** (*bool*) – Whether to use distributed training. Default: False.
- **validate** (*bool*) – Whether to do evaluation. Default: False.
- **timestamp** (*str | None*) – Local time for runner. Default: None.
- **meta** (*dict | None*) – Meta dict to record some important information. Default: None

`mmpose.apis.vis_3d_mesh_result(model, result, img=None, show=False, out_file=None)`

Visualize the 3D mesh estimation results.

Parameters

- **model** (*nn.Module*) – The loaded model.
- **result** (*list[dict]*) – 3D mesh estimation results.

`mmpose.apis.vis_3d_pose_result(model, result, img=None, dataset='Body3DH36MDataset', dataset_info=None, kpt_score_thr=0.3, radius=8, thickness=2, vis_height=400, num_instances=-1, axis_azimuth=70, show=False, out_file=None)`

Visualize the 3D pose estimation results.

Parameters

- **model** (*nn.Module*) – The loaded model.
- **result** (*list[dict]*) –

```
mmpose.apis.vis_pose_result(model, img, result, radius=4, thickness=1, kpt_score_thr=0.3,  
                             bbox_color='green', dataset='TopDownCocoDataset', dataset_info=None,  
                             show=False, out_file=None)
```

Visualize the detection results on the image.

Parameters

- **model** (*nn.Module*) – The loaded detector.
- **img** (*str* | *np.ndarray*) – Image filename or loaded image.
- **result** (*list[dict]*) – The results to draw over *img* (bbox_result, pose_result).
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **kpt_score_thr** (*float*) – The threshold to visualize the keypoints.
- **skeleton** (*list[tuple()]*) – Default None.
- **show** (*bool*) – Whether to show the image. Default True.
- **out_file** (*str/None*) – The filename of the output visualization image.

```
mmpose.apis.vis_pose_tracking_result(model, img, result, radius=4, thickness=1, kpt_score_thr=0.3,  
                                     dataset='TopDownCocoDataset', dataset_info=None, show=False,  
                                     out_file=None)
```

Visualize the pose tracking results on the image.

Parameters

- **model** (*nn.Module*) – The loaded detector.
- **img** (*str* | *np.ndarray*) – Image filename or loaded image.
- **result** (*list[dict]*) – The results to draw over *img* (bbox_result, pose_result).
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **kpt_score_thr** (*float*) – The threshold to visualize the keypoints.
- **skeleton** (*list[tuple]*) – Default None.
- **show** (*bool*) – Whether to show the image. Default True.
- **out_file** (*str/None*) – The filename of the output visualization image.

MMPOSE.APIS.WEBCAM

MMPose Webcam API: Tools to build simple interactive webcam applications and demos

- *Executor*
- *Nodes*
 - *Base Nodes*
 - *Model Nodes*
 - *Visualizer Nodes*
 - *Helper Nodes*
- *Utils*
 - *Buffer and Message*
 - *Pose*
 - *Event*
 - *Misc*

47.1 Executor

WebcamExecutor

The interface to build and execute webcam applications from configs.

47.1.1 mmpose.apis.webcam.WebcamExecutor

```
class mmpose.apis.webcam.WebcamExecutor(nodes: List[Dict], name: str = 'MMPose Webcam App',
                                         camera_id: Union[int, str] = 0, camera_max_fps: int = 30,
                                         camera_frame_shape: Optional[Tuple[int, int]] = None,
                                         synchronous: bool = False, buffer_sizes: Optional[Dict[str, int]]
                                         = None)
```

The interface to build and execute webcam applications from configs.

Parameters

- **nodes** (*list[dict]*) – Node configs. See `webcam.nodes.Node` for details

- **name** (*str*) – Executor name. Default: ‘MMPose Webcam App’.
- **camera_id** (*int* / *str*) – The camera ID (usually the ID of the default camera is 0). Alternatively a file path or a URL can be given to load from a video or image file.
- **camera_frame_shape** (*tuple*, *optional*) – Set the frame shape of the camera in (width, height). If not given, the default frame shape will be used. This argument is only valid when using a camera as the input source. Default: *None*
- **camera_max_fps** (*int*) – Video reading maximum FPS. Default: 30
- **buffer_sizes** (*dict*, *optional*) – A dict to specify buffer sizes. The key is the buffer name and the value is the buffer size. Default: *None*

Example::

```
>>> cfg = dict(
>>>     name='Test Webcam',
>>>     camera_id=0,
>>>     camera_max_fps=30,
>>>     nodes=[
>>>         dict(
>>>             type='MonitorNode',
>>>             name='monitor',
>>>             enable_key='m',
>>>             enable=False,
>>>             input_buffer='_frame_',
>>>             output_buffer='display'),
>>>         dict(
>>>             type='RecorderNode',
>>>             name='recorder',
>>>             out_video_file='webcam_output.mp4',
>>>             input_buffer='display',
>>>             output_buffer='_display_')
>>>     ])
```

```
>>> executor = WebcamExecutor(**cfg)
```

__init__ (*nodes: List[Dict]*, *name: str* = 'MMPose Webcam App', *camera_id: Union[int, str]* = 0, *camera_max_fps: int* = 30, *camera_frame_shape: Optional[Tuple[int, int]]* = *None*, *synchronous: bool* = *False*, *buffer_sizes: Optional[Dict[str, int]]* = *None*)

Initialize self. See help(type(self)) for accurate signature.

Methods

<code>__init__</code> (<i>nodes</i> [, <i>name</i> , <i>camera_id</i> , ...])	Initialize self.
<code>run</code> ()	Start the executor.

47.2 Nodes

47.2.1 Base Nodes

<i>Node</i>	Base class for node, which is the interface of basic function module.
<i>BaseVisualizerNode</i>	Base class for nodes whose function is to create visual effects, like visualizing model predictions, showing graphics or showing text messages.

Node

```
class mmpose.apis.webcam.nodes.Node(name: str, enable_key: Optional[Union[str, int]] = None, max_fps:
    int = 30, input_check_interval: float = 0.01, enable: bool = True,
    daemon: bool = False, multi_input: bool = False)
```

Base class for node, which is the interface of basic function module.

Node inherits `threading.Thread`. All subclasses should override following methods:

- `process()`
- `bypass()` (optional)

Parameters

- **name** (*str*) – The node name (also thread name)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If `enable_key` is set, the `bypass()` method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: ‘q’, ‘Q’ and 27 are used for exiting. Default: `None`
- **max_fps** (*int*) – Maximum FPS of the node. This is to avoid the node running unrestrictedly and causing large resource consuming. Default: 30
- **input_check_interval** (*float*) – Minimum interval (in millisecond) between checking if input is ready. Default: 0.001
- **enable** (*bool*) – Default enable/disable status. Default: `True`
- **daemon** (*bool*) – Whether node is a daemon. Default: `True`
- **multi_input** (*bool*) – Whether load all messages in buffer. If `False`, only one message will be loaded each time. Default: `False`

bypass(*input_msgs: Dict[str, mmpose.apis.webcam.utils.message.Message]*) →
Optional[*mmpose.apis.webcam.utils.message.Message*]
The method that defines the node behavior when disabled.

Note that a node must override this method if it has `enable_key`. This method has the same signature as `process()`.

Parameters **input_msgs** (*dict[str, Message]*) – The input data collected from the buffers. For each item, the key is the `input_name` of the registered input buffer, and the value is a `Message` instance fetched from the buffer (or `None` if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

on_exit()

This method will be invoked on event `_exit_`.

Subclasses should override this method to specifying the exiting behavior.

abstract process(*input_msgs: Dict[str, mmpose.apis.webcam.utils.message.Message]*) →
Optional[*mmpose.apis.webcam.utils.message.Message*]

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters **input_msgs** (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

register_event(*event_name: str, is_keyboard: bool = False, handler_func: Optional[Callable] = None*)

Register an event. All events used in the node need to be registered in `__init__`. If a callable handler is given, a thread will be create to listen and handle the event when the node starts.

Parameters

- **Args** –
- **event_name** (*str/int*) – The event name. If `is_keyboard==True`, *event_name* should be a *str* (as char) or an *int* (as ascii)
- **is_keyboard** (*bool*) – Indicate whether it is an keyboard event. If `True`, the argument *event_name* will be regarded as a key indicator.
- **handler_func** (*callable, optional*) – The event handler function, which should be a collable object with no arguments or return values. Default: `None`.

register_input_buffer(*buffer_name: str, input_name: str, trigger: bool = False*)

Register an input buffer, so that Node can automatically check if data is ready, fetch data from the buffers and format the inputs to feed into *process* method.

The subclass of Node should invoke *register_input_buffer* in its `__init__` method. This method can be invoked multiple times to register multiple input buffers.

Parameters

- **buffer_name** (*str*) – The name of the buffer
- **input_name** (*str*) – The name of the fetched message from the corresponding buffer
- **trigger** (*bool*) – An trigger input means the node will wait until the input is ready before processing. Otherwise, an inessential input will not block the processing, instead a `None` will be fetched if the buffer is not ready.

register_output_buffer(*buffer_name: Union[str, List[str]]*)

Register one or multiple output buffers, so that the Node can automatically send the output of the *process* method to these buffers.

The subclass of Node should invoke *register_output_buffer* in its `__init__` method.

Parameters **buffer_name** (*str/list*) – The name(s) of the output buffer(s).

run()

Method representing the Node's activity.

This method override the standard `run()` method of `Thread`. Subclasses of `Node` should not override this method in subclasses.

set_executor(executor)

Assign the node to an executor so the node can access the buffers and event manager of the executor.

This method should be invoked by the executor instance.

Parameters `executor` (`WebcamExecutor`) – The executor to hold the node

BaseVisualizerNode

```
class mmpose.apis.webcam.nodes.BaseVisualizerNode(name: str, input_buffer: str, output_buffer:
                                                Union[str, List[str]], enable_key:
                                                Optional[Union[str, int]] = None, enable: bool =
                                                True)
```

Base class for nodes whose function is to create visual effects, like visualizing model predictions, showing graphics or showing text messages.

All subclass should implement the method `draw()`.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str* / *list*) – The name(s) of the output buffer(s).
- **enable_key** (*str/int*, *optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If `enable_key` is set, the `bypass()` method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: `None`
- **enable** (*bool*) – Default enable/disable status. Default: `True`

bypass(*input_msgs: Dict[str, mmpose.apis.webcam.utils.message.Message]*) → *Optional[mmpose.apis.webcam.utils.message.Message]*

The method that defines the node behavior when disabled.

Note that a node must override this method if it has `enable_key`. This method has the same signature as `process()`.

Parameters `input_msgs` (*dict[str, Message]*) – The input data collected from the buffers. For each item, the key is the `input_name` of the registered input buffer, and the value is a `Message` instance fetched from the buffer (or `None` if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

abstract draw(*input_msg: mmpose.apis.webcam.utils.message.FrameMessage*) → *numpy.ndarray*

Draw on the frame image of the input `FrameMessage`.

Parameters `input_msg` (*FrameMessage*) – The message of the frame to draw on

Returns The processed image.

Return type *np.array*

process(*input_msgs*: Dict[str, [mmpose.apis.webcam.utils.message.Message](#)]) →
Optional[[mmpose.apis.webcam.utils.message.Message](#)]

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters **input_msgs** (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type [Message](#)

47.2.2 Model Nodes

DetectorNode	Detect objects from the frame image using MMDetection model.
TopDownPoseEstimatorNode	Perform top-down pose estimation using MMPose model.
PoseTrackerNode	Perform object detection and top-down pose estimation.

DetectorNode

```
class mmpose.apis.webcam.nodes.DetectorNode(name: str, model_config: str, model_checkpoint: str,
                                             input_buffer: str, output_buffer: Union[str, List[str]],
                                             enable_key: Optional[Union[str, int]] = None, enable:
                                             bool = True, device: str = 'cuda:0', bbox_thr: float = 0.5,
                                             multi_input: bool = False)
```

Detect objects from the frame image using MMDetection model.

Note that MMDetection is required for this node. Please refer to [MMDetection documentation](#) for the installation guide.

Parameters

- **name** (*str*) – The node name (also thread name)
- **model_cfg** (*str*) – The model config file
- **model_checkpoint** (*str*) – The model checkpoint file
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If *enable_key* is set, the *bypass()* method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: None
- **enable** (*bool*) – Default enable/disable status. Default: True
- **device** (*str*) – Specify the device to hold model weights and inference the model. Default: 'cuda:0'
- **bbox_thr** (*float*) – Set a threshold to filter out objects with low bbox scores. Default: 0.5

- **multi_input** (*bool*) – Whether load all frames in input buffer. If True, all frames in buffer will be loaded and stacked. The latest frame is used to detect objects of interest. Default: False

Example::

```
>>> cfg = dict(
...     type='DetectorNode',
...     name='detector',
...     model_config='demo/mmdetection_cfg/'
...     'ssdlite_mobilenetv2_scratch_600e_coco.py',
...     model_checkpoint='https://download.openmmlab.com'
...     '/mmdetection/v2.0/ssd/'
...     'ssdlite_mobilenetv2_scratch_600e_coco/ssdlite_mobilenetv2_'
...     'scratch_600e_coco_20210629_110627-974d9307.pth',
...     # `_input` is an executor-reserved buffer
...     input_buffer='_input_',
...     output_buffer='det_result')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

bypass(*input_msgs*)

The method that defines the node behavior when disabled.

Note that a node must override this method if it has *enable_key*. This method has the same signature as `process()`.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

TopDownPoseEstimatorNode

```
class mmpose.apis.webcam.nodes.TopDownPoseEstimatorNode(name: str, model_config: str,
                                                         model_checkpoint: str, input_buffer: str,
                                                         output_buffer: Union[str, List[str]],
                                                         enable_key: Optional[Union[str, int]] =
                                                         None, enable: bool = True, device: str =
                                                         'cuda:0', class_ids: Optional[List[int]] =
                                                         None, labels: Optional[List[str]] = None,
                                                         bbox_thr: float = 0.5, smooth: bool =
                                                         False, smooth_filter_cfg: str =
                                                         'configs/_base_/filters/one_euro.py')
```

Perform top-down pose estimation using MMPose model.

The node should be placed after an object detection node.

Parameters

- **name** (*str*) – The node name (also thread name)
- **model_cfg** (*str*) – The model config file
- **model_checkpoint** (*str*) – The model checkpoint file
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If `enable_key` is set, the `bypass()` method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: `None`
- **enable** (*bool*) – Default enable/disable status. Default: `True`
- **device** (*str*) – Specify the device to hold model weights and inference the model. Default: `'cuda:0'`
- **class_ids** (*list[int], optional*) – Specify the object category indices to apply pose estimation. If both `class_ids` and `labels` are given, `labels` will be ignored. If neither is given, pose estimation will be applied for all objects. Default: `None`
- **labels** (*list[str], optional*) – Specify the object category names to apply pose estimation. See also `class_ids`. Default: `None`
- **bbox_thr** (*float*) – Set a threshold to filter out objects with low bbox scores. Default: `0.5`
- **smooth** (*bool*) – If set to `True`, a `Smoother` will be used to refine the pose estimation result. Default: `True`
- **smooth_filter_cfg** (*str*) – The filter config path to build the smoother. Only valid when `smooth==True`. By default, `OneEuro` filter will be used.

Example::

```
>>> cfg = dict(
...     type='TopDownPoseEstimatorNode',
...     name='human pose estimator',
...     model_config='configs/wholebody/2d_kpt_sview_rgb_img/'
...     'topdown_heatmap/coco-wholebody/')
```

(continues on next page)

(continued from previous page)

```

...     'vipnas_mbv3_coco_wholebody_256x192_dark.py',
...     model_checkpoint='https://download.openmmlab.com/mmpose/'
...     'top_down/vipnas/vipnas_mbv3_coco_wholebody_256x192_dark'
...     '-e2158108_20211205.pth',
...     labels=['person'],
...     smooth=True,
...     input_buffer='det_result',
...     output_buffer='human_pose')

```

```

>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)

```

bypass(*input_msgs*)

The method that defines the node behavior when disabled.

Note that a node must override this method if it has *enable_key*. This method has the same signature as `process()`.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

PoseTrackerNode

```

class mmpose.apis.webcam.nodes.PoseTrackerNode(name: str, det_model_config: str,
                                                det_model_checkpoint: str, pose_model_config: str,
                                                pose_model_checkpoint: str, input_buffer: str,
                                                output_buffer: Union[str, List[str]], enable_key:
                                                Optional[Union[str, int]] = None, enable: bool = True,
                                                device: str = 'cuda:0', det_interval: int = 1, class_ids:
                                                Optional[List] = None, labels: Optional[List] = None,
                                                bbox_thr: float = 0.5, kpt2bbox_cfg: Optional[dict] =
                                                None, smooth: bool = False, smooth_filter_cfg: str =
                                                'configs/_base_/filters/one_euro.py')

```

Perform object detection and top-down pose estimation. Only detect objects every few frames, and use the pose estimation results to track the object at interval.

Note that MMDetection is required for this node. Please refer to [MMDetection documentation](#) for the installation guide.

Parameters

- **name** (*str*) – The node name (also thread name)
- **det_model_cfg** (*str*) – The config file of the detection model
- **det_model_checkpoint** (*str*) – The checkpoint file of the detection model
- **pose_model_cfg** (*str*) – The config file of the pose estimation model
- **pose_model_checkpoint** (*str*) – The checkpoint file of the pose estimation model
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If enable_key is set, the bypass() method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: None
- **enable** (*bool*) – Default enable/disable status. Default: True
- **device** (*str*) – Specify the device to hold model weights and inference the model. Default: 'cuda:0'
- **det_interval** (*int*) – Set the detection interval in frames. For example, det_interval==10 means inference the detection model every 10 frames. Default: 1
- **class_ids** (*list[int], optional*) – Specify the object category indices to apply pose estimation. If both class_ids and labels are given, labels will be ignored. If neither is given, pose estimation will be applied for all objects. Default: None
- **labels** (*list[str], optional*) – Specify the object category names to apply pose estimation. See also class_ids. Default: None
- **bbox_thr** (*float*) – Set a threshold to filter out objects with low bbox scores. Default: 0.5
- **kpt2bbox_cfg** (*dict, optional*) – Configure the process to get object bbox from its keypoints during tracking. Specifically, the bbox is obtained from the minimal outer rectangle of the keypoints with following configurable arguments: 'scale', the coefficient to expand the keypoint outer rectangle, defaults to 1.5; 'kpt_thr': a threshold to filter out low-scored keypoint, defaults to 0.3. See self.default_kpt2bbox_cfg for details
- **smooth** (*bool*) – If set to True, a Smoother will be used to refine the pose estimation result. Default: True
- **smooth_filter_cfg** (*str*) – The filter config path to build the smoother. Only valid when smooth==True. Default to use an OneEuro filter

Example::

```
>>> cfg = dict(
...     type='PoseTrackerNode',
...     name='pose tracker',
...     det_model_config='demo/mmdetection_cfg/'
...     'ssdlite_mobilenetv2_scratch_600e_coco.py',
```

(continues on next page)

(continued from previous page)

```

...   det_model_checkpoint='https://download.openmmlab.com'
...   '/mmdetection/v2.0/ssd/'
...   'ssdlite_mobilenetv2_scratch_600e_coco/ssdlite_mobilenetv2_'
...   'scratch_600e_coco_20210629_110627-974d9307.pth',
...   pose_model_config='configs/wholebody/2d_kpt_sview_rgb_img/'
...   'topdown_heatmap/coco-wholebody/'
...   'vipnas_mbv3_coco_wholebody_256x192_dark.py',
...   pose_model_checkpoint='https://download.openmmlab.com/mmpose/'
...   'top_down/vipnas/vipnas_mbv3_coco_wholebody_256x192_dark'
...   '-e2158108_20211205.pth',
...   det_interval=10,
...   labels=['person'],
...   smooth=True,
...   device='cuda:0',
...   # `_input` is an executor-reserved buffer
...   input_buffer='_input_',
...   output_buffer='human_pose')

```

```

>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)

```

bypass(*input_msgs*)

The method that defines the node behavior when disabled.

Note that a node must override this method if it has *enable_key*. This method has the same signature as `process()`.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

47.2.3 Visualizer Nodes

<i>ObjectVisualizerNode</i>	Visualize the bounding box and keypoints of objects.
<i>NoticeBoardNode</i>	Show text messages in the frame.
<i>SunglassesEffectNode</i>	Apply sunglasses effect (draw sunglasses at the facial area) to the objects with eye keypoints in the frame.
<i>BigeyeEffectNode</i>	Apply big-eye effect to the objects with eye keypoints in the frame.

ObjectVisualizerNode

```
class mmpose.apis.webcam.nodes.ObjectVisualizerNode(name: str, input_buffer: str, output_buffer:
                                                    Union[str, List[str]], enable_key:
                                                    Optional[Union[str, int]] = None, enable: bool
                                                    = True, show_bbox: bool = True,
                                                    show_keypoint: bool = True,
                                                    must_have_keypoint: bool = False, kpt_thr: float
                                                    = 0.3, radius: int = 4, thickness: int = 2,
                                                    bbox_color: Optional[Union[str, Tuple, Dict]] =
                                                    None)
```

Visualize the bounding box and keypoints of objects.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If **enable_key** is set, the **bypass()** method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: ‘q’, ‘Q’ and 27 are used for exiting. Default: **None**
- **enable** (*bool*) – Default enable/disable status. Default: **True**
- **show_bbox** (*bool*) – Set **True** to show the bboxes of detection objects. Default: **True**
- **show_keypoint** (*bool*) – Set **True** to show the pose estimation results. Default: **True**
- **must_have_bbox** (*bool*) – Only show objects with keypoints. Default: **False**
- **kpt_thr** (*float*) – The threshold of keypoint score. Default: 0.3
- **radius** (*int*) – The radius of keypoint. Default: 4
- **thickness** (*int*) – The thickness of skeleton. Default: 2
- **bbox_color** (*str/tuple/dict*) – The color of bboxes. If a single color is given (a str like ‘green’ or a BGR tuple like (0, 255, 0)), it will be used for all bboxes. If a dict is given, it will be used as a map from class labels to bbox colors. If not given, a default color map will be used. Default: **None**

Example::

```
>>> cfg = dict(
...     type='ObjectVisualizerNode',
...     name='object visualizer',
...     enable_key='v',
...     enable=True,
...     show_bbox=True,
...     must_have_keypoint=False,
...     show_keypoint=True,
...     input_buffer='frame',
...     output_buffer='vis')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

draw(*input_msg*: `mmpose.apis.webcam.utils.message.FrameMessage`) → `numpy.ndarray`
 Draw on the frame image of the input `FrameMessage`.

Parameters *input_msg* (`FrameMessage`) – The message of the frame to draw on

Returns The processed image.

Return type `np.array`

NoticeBoardNode

```
class mmpose.apis.webcam.nodes.NoticeBoardNode(name: str, input_buffer: str, output_buffer: Union[str,
List[str]], enable_key: Optional[Union[str, int]] =
None, enable: bool = True, content_lines:
Optional[List[str]] = None, x_offset: int = 20, y_offset:
int = 20, y_delta: int = 15, text_color: Union[str,
Tuple[int, int, int]] = 'black', background_color:
Union[str, Tuple[int, int, int]] = (255, 183, 0),
text_scale: float = 0.4)
```

Show text messages in the frame.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If `enable_key` is set, the `bypass()` method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: `None`
- **enable** (*bool*) – Default enable/disable status. Default: `True`
- **content_lines** (*list[str], optional*) – The lines of text message to show in the frame. If not given, a default message will be shown. Default: `None`
- **x_offset** (*int*) – The position of the notice board's left border in pixels. Default: 20
- **y_offset** (*int*) – The position of the notice board's top border in pixels. Default: 20

- **y_delta** (*int*) – The line height in pixels. Default: 15
- **text_color** (*str/tuple*) – The font color represented in a color name or a BGR tuple. Default: 'black'
- **background_color** (*str/tuple*) – The background color represented in a color name or a BGR tuple. Default: (255, 183, 0)
- **text_scale** (*float*) – The font scale factor that is multiplied by the base size. Default: 0.4

Example::

```
>>> cfg = dict(
...     type='NoticeBoardNode',
...     name='instruction',
...     enable_key='h',
...     enable=True,
...     input_buffer='vis_bigeye',
...     output_buffer='vis_notice',
...     content_lines=[
...         'This is a demo for pose visualization and simple image '
...         'effects. Have fun!', '', 'Hot-keys:',
...         '"v": Pose estimation result visualization',
...         '"s": Sunglasses effect B-)', '"b": Big-eye effect @_@',
...         '"h": Show help information',
...         '"m": Show diagnostic information', '"q": Exit'
...     ],
... )
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

draw(*input_msg*: mmpose.apis.webcam.utils.message.FrameMessage) → numpy.ndarray

Draw on the frame image of the input FrameMessage.

Parameters **input_msg** (FrameMessage) – The message of the frame to draw on

Returns The processed image.

Return type np.array

SunglassesEffectNode

```
class mmpose.apis.webcam.nodes.SunglassesEffectNode(name: str, input_buffer: str, output_buffer:
    Union[str, List[str]], enable_key:
    Optional[Union[str, int]] = None, enable: bool
    = True, kpt_thr: float = 0.5, resource_img_path:
    Optional[str] = None)
```

Apply sunglasses effect (draw sunglasses at the facial area) to the objects with eye keypoints in the frame.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)

- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note:
 1. If **enable_key** is set, the **bypass method** need to be overridden to define the node behavior when disabled
 2. Some hot-key has been use for particular use. For example: 'q', 'Q' and 27 are used for quit
 Default: None
- **enable** (*bool*) – Default enable/disable status. Default: True.
- **kpt_thr** (*float*) – The score threshold of valid keypoints. Default: 0.5
- **resource_img_path** (*str, optional*) – The resource image path or url. The image should be a pair of sunglasses with white background. If not specified, the url of a default image will be used. See `SunglassesNode.default_resource_img_path`. Default: None

Example::

```
>>> cfg = dict(
...     type='SunglassesEffectNode',
...     name='sunglasses',
...     enable_key='s',
...     enable=False,
...     input_buffer='vis',
...     output_buffer='vis_sunglasses')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

apply_sunglasses_effect (*canvas: numpy.ndarray, objects: List[Dict], left_eye_index: int, right_eye_index: int*) → *numpy.ndarray*

Apply sunglasses effect.

Parameters

- **canvas** (*np.ndarray*) – The image to apply the effect
- **objects** (*list[dict]*) – The object list with keypoints - “keypoints” ([K,3]): keypoints in [x, y, score]
- **left_eye_index** (*int*) – Keypoint index of the left eye
- **right_eye_index** (*int*) – Keypoint index of the right eye

Returns Processed image

Return type *np.ndarray*

draw (*input_msg*)

Draw on the frame image of the input `FrameMessage`.

Parameters **input_msg** (*FrameMessage*) – The message of the frame to draw on

Returns The processed image.

Return type *np.array*

BigeyeEffectNode

class `mmpose.apis.webcam.nodes.BigeyeEffectNode`(*name: str, input_buffer: str, output_buffer: Union[str, List[str]], enable_key: Optional[Union[str, int]] = None, enable: bool = True, kpt_thr: float = 0.5*)

Apply big-eye effect to the objects with eye keypoints in the frame.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If `enable_key` is set, the `bypass()` method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: ‘q’, ‘Q’ and 27 are used for exiting. Default: `None`
- **enable** (*bool*) – Default enable/disable status. Default: `True`
- **kpt_thr** (*float*) – The score threshold of valid keypoints. Default: `0.5`

Example::

```
>>> cfg = dict(
...     type='SunglassesEffectNode',
...     name='sunglasses',
...     enable_key='s',
...     enable=False,
...     input_buffer='vis',
...     output_buffer='vis_sunglasses')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

apply_bigeye_effect(*canvas: numpy.ndarray, objects: List[Dict], left_eye_index: int, right_eye_index: int*) → *numpy.ndarray*

Apply big-eye effect.

Parameters

- **canvas** (*np.ndarray*) – The image to apply the effect
- **objects** (*list[dict]*) – The object list with bbox and keypoints - “bbox” ([K, 4(or 5)]):
bbox in [x1, y1, x2, y2, (score)] - “keypoints” ([K,3]): keypoints in [x, y, score]
- **left_eye_index** (*int*) – Keypoint index of left eye
- **right_eye_index** (*int*) – Keypoint index of right eye

Returns Processed image.

Return type *np.ndarray*

draw(*input_msg*)

Draw on the frame image of the input `FrameMessage`.

Parameters **input_msg** (*FrameMessage*) – The message of the frame to draw on

Returns The processed image.

Return type np.array

47.2.4 Helper Nodes

<i>ObjectAssignerNode</i>	Assign the object information to the frame message.
<i>MonitorNode</i>	Show diagnostic information.
<i>RecorderNode</i>	Record the video frames into a local file.

ObjectAssignerNode

class `mmpose.apis.webcam.nodes.ObjectAssignerNode`(*name: str, frame_buffer: str, object_buffer: str, output_buffer: Union[str, List[str]]*)

Assign the object information to the frame message.

ObjectAssignerNode enables asynchronous processing of model inference and video I/O, so the video will be captured and displayed smoothly regardless of the model inference speed. Specifically, *ObjectAssignerNode* takes messages from both model branch and video I/O branch as its input, indicated as “object message” and “frame message” respectively. When an object message arrives it will update the latest object information; and when a frame message arrives, it will be assigned with the latest object information and output.

Specially, if the webcam executor is set to synchronous mode, the behavior of *ObjectAssignerNode* will be different: When an object message arrives, it will trigger an output of itself; and the frame messages will be ignored.

Parameters

- **name** (*str*) – The node name (also thread name)
- **frame_buffer** (*str*) – Buffer name for frame messages
- **object_buffer** (*str*) – Buffer name for object messages
- **output_buffer** (*str*) – The name(s) of the output buffer(s)

Example::

```
>>> cfg =dict(
...     type='ObjectAssignerNode',
...     name='object assigner',
...     frame_buffer='_frame_',
...     # `_frame_` is an executor-reserved buffer
...     object_buffer='animal_pose',
...     output_buffer='frame')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters **input_msgs** (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

set_executor(*executor*)

Assign the node to an executor so the node can access the buffers and event manager of the executor.

This method should be invoked by the executor instance.

Parameters **executor** (WebcamExecutor) – The executor to hold the node

MonitorNode

```
class mmpose.apis.webcam.nodes.MonitorNode(name: str, input_buffer: str, output_buffer: Union[str,
List[str]], enable_key: Optional[Union[str, int]] = None,
enable: bool = False, x_offset=20, y_offset=20, y_delta=15,
text_color='black', background_color=(255, 183, 0),
text_scale=0.4, ignore_items: Optional[List[str]] = None)
```

Show diagnostic information.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **enable_key** (*str/int, optional*) – Set a hot-key to toggle enable/disable of the node. If an int value is given, it will be treated as an ascii code of a key. Please note: (1) If *enable_key* is set, the *bypass()* method need to be overridden to define the node behavior when disabled; (2) Some hot-keys are reserved for particular use. For example: 'q', 'Q' and 27 are used for exiting. Default: None
- **enable** (*bool*) – Default enable/disable status. Default: True
- **x_offset** (*int*) – The position of the text box's left border in pixels. Default: 20
- **y_offset** (*int*) – The position of the text box's top border in pixels. Default: 20
- **y_delta** (*int*) – The line height in pixels. Default: 15
- **text_color** (*str/tuple*) – The font color represented in a color name or a BGR tuple. Default: 'black'
- **background_color** (*str/tuple*) – The background color represented in a color name or a BGR tuple. Default: (255, 183, 0)
- **text_scale** (*float*) – The font scale factor that is multiplied by the base size. Default: 0.4
- **ignore_items** (*list[str], optional*) – Specify the node information items that will not be shown. See *MonitorNode._default_ignore_items* for the default setting.

Example::

```
>>> cfg = dict(
...     type='MonitorNode',
...     name='monitor',
...     enable_key='m',
...     enable=False,
...     input_buffer='vis_notice',
...     output_buffer='display')
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

bypass(*input_msgs*)

The method that defines the node behavior when disabled.

Note that a node must override this method if it has *enable_key*. This method has the same signature as `process()`.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters *input_msgs* (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

RecorderNode

```
class mmpose.apis.webcam.nodes.RecorderNode(name: str, input_buffer: str, output_buffer: Union[str,
List[str]], out_video_file: str, out_video_fps: int = 30,
out_video_codec: str = 'mp4v', buffer_size: int = 30,
enable: bool = True)
```

Record the video frames into a local file.

RecorderNode uses OpenCV backend to record the video. Recording is performed in a separate thread to avoid blocking the data stream. A buffer queue is used to cached the arrived frame images.

Parameters

- **name** (*str*) – The node name (also thread name)
- **input_buffer** (*str*) – The name of the input buffer
- **output_buffer** (*str/list*) – The name(s) of the output buffer(s)
- **out_video_file** (*str*) – The path of the output video file

- **out_video_fps** (*int*) – The frame rate of the output video. Default: 30
- **out_video_codec** (*str*) – The codec of the output video. Default: 'mp4v'
- **buffer_size** (*int*) – Size of the buffer queue that caches the arrived frame images.
- **enable** (*bool*) – Default enable/disable status. Default: True.

Example::

```
>>> cfg = dict(
...     type='RecorderNode',
...     name='recorder',
...     out_video_file='webcam_demo.mp4',
...     input_buffer='display',
...     output_buffer='_display_'
...     # `_display_` is an executor-reserved buffer
... )
```

```
>>> from mmpose.apis.webcam.nodes import NODES
>>> node = NODES.build(cfg)
```

on_exit()

This method will be invoked on event `_exit_`.

Subclasses should override this method to specifying the exiting behavior.

process(*input_msgs*)

The method that implements the function of the node.

This method will be invoked when the node is enabled and the input data is ready. All subclasses of Node should override this method.

Parameters **input_msgs** (dict[str, Message]) – The input data collected from the buffers. For each item, the key is the *input_name* of the registered input buffer, and the value is a Message instance fetched from the buffer (or None if the buffer is non-trigger and not ready).

Returns The output message of the node which will be send to all registered output buffers.

Return type *Message*

47.3 Utils

47.3.1 Buffer and Message

<i>BufferManager</i>	A helper class to manage multiple buffers.
<i>Message</i>	Message base class.
<i>FrameMessage</i>	The message to store information of a video frame.
<i>VideoEndingMessage</i>	The special message to indicate the ending of the input video.

mmpose.apis.webcam.utils.BufferManager

```
class mmpose.apis.webcam.utils.BufferManager(buffer_type: type = <class
'mmpose.apis.webcam.utils.buffer.Buffer'>, buffers:
Optional[Dict] = None)
```

A helper class to manage multiple buffers.

Parameters

- **buffer_type** (*type*) – The class to build buffer instances. Default: `mmpose.apis.webcam.utils.buffer.Buffer`.
- **buffers** (*dict, optional*) – Create [BufferManager](#) from existing buffers. Each item should a buffer name and the buffer. If not given, an empty buffer manager will be create. Default: `None`

```
__init__(buffer_type: type = <class 'mmpose.apis.webcam.utils.buffer.Buffer'>, buffers: Optional[Dict] =
None)
```

Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__([buffer_type, buffers])</code>	Initialize self.
<code>get(name[, block, timeout])</code>	Remove an return an item from the specified buffer.
<code>get_info()</code>	Returns the information of all buffers in the manager.
<code>get_sub_manager(buffer_names)</code>	Return a BufferManager instance that covers a subset of the buffers in the parent.
<code>is_empty(name)</code>	Check if a buffer is empty.
<code>is_full(name)</code>	Check if a buffer is full.
<code>put(name, item[, block, timeout])</code>	Put an item into specified buffer.
<code>put_force(name, item)</code>	Force to put an item into specified buffer.
<code>register_buffer(name[, maxsize])</code>	Register a buffer.

mmpose.apis.webcam.utils.Message

```
class mmpose.apis.webcam.utils.Message(msg: str = "", data: Optional[Dict] = None)
```

Message base class.

All message class should inherit this class. The basic use of a Message instance is to carry a piece of text message (`self.msg`) and a dict that stores structured data (`self.data`), e.g. frame image, model prediction, et al.

A message may also hold route information, which is composed of information of all nodes the message has passed through.

Parameters

- **msg** (*str*) – The text message.
- **data** (*dict, optional*) – The structured data.

```
__init__(msg: str = "", data: Optional[Dict] = None)
```

Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__([msg, data])</code>	Initialize self.
<code>get_route_info()</code>	
<code>merge_route_info(route_info)</code>	Merge the given route information into the original one of the message.
<code>set_route_info(route_info)</code>	Directly set the entire route information.
<code>update_route_info([node, node_name, ...])</code>	Append new node information to the route information.

`mmpose.apis.webcam.utils.FrameMessage`

class `mmpose.apis.webcam.utils.FrameMessage(img)`

The message to store information of a video frame.

`__init__(img)`

Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__(img)</code>	Initialize self.
<code>get_image()</code>	Get the frame image.
<code>get_objects([obj_filter])</code>	Get object information from the frame data.
<code>get_route_info()</code>	
<code>merge_route_info(route_info)</code>	Merge the given route information into the original one of the message.
<code>set_image(img)</code>	Set the frame image to the message.
<code>set_objects(objects)</code>	Set the object information.
<code>set_route_info(route_info)</code>	Directly set the entire route information.
<code>update_objects(objects)</code>	Update object information.
<code>update_route_info([node, node_name, ...])</code>	Append new node information to the route information.

`mmpose.apis.webcam.utils.VideoEndingMessage`

class `mmpose.apis.webcam.utils.VideoEndingMessage(msg: str = "", data: Optional[Dict] = None)`

The special message to indicate the ending of the input video.

`__init__(msg: str = "", data: Optional[Dict] = None)`

Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__([msg, data])</code>	Initialize self.
<code>get_route_info()</code>	
<code>merge_route_info(route_info)</code>	Merge the given route information into the original one of the message.
<code>set_route_info(route_info)</code>	Directly set the entire route information.
<code>update_route_info([node, node_name, ...])</code>	Append new node information to the route information.

47.3.2 Pose

<code>get_eye_keypoint_ids</code>	A helper function to get the keypoint indices of left and right eyes from the model config.
<code>get_face_keypoint_ids</code>	A helper function to get the keypoint indices of the face from the model config.
<code>get_hand_keypoint_ids</code>	A helper function to get the keypoint indices of left and right hand from the model config.
<code>get_mouth_keypoint_ids</code>	A helper function to get the mouth keypoint index from the model config.
<code>get_wrist_keypoint_ids</code>	A helper function to get the keypoint indices of left and right wrists from the model config.

mmpose.apis.webcam.utils.get_eye_keypoint_ids

`mmpose.apis.webcam.utils.get_eye_keypoint_ids(model_cfg: mmcv.utils.config.Config) → Tuple[int, int]`
 A helper function to get the keypoint indices of left and right eyes from the model config.

Parameters `model_cfg (Config)` – mmpose model config

Returns The keypoint indices of left eye and right eye.

Return type `tuple[int, int]`

mmpose.apis.webcam.utils.get_face_keypoint_ids

`mmpose.apis.webcam.utils.get_face_keypoint_ids(model_cfg: mmcv.utils.config.Config) → List`
 A helper function to get the keypoint indices of the face from the model config.

Parameters `model_cfg (Config)` – pose model config.

Returns face keypoint indices. The length depends on the dataset.

Return type `list[int]`

mmpose.apis.webcam.utils.get_hand_keypoint_ids

`mmpose.apis.webcam.utils.get_hand_keypoint_ids(model_cfg: mmcv.utils.config.Config) → List[int]`

A helper function to get the keypoint indices of left and right hand from the model config.

Parameters `model_cfg (Config)` – pose model config.

Returns hand keypoint indices. The length depends on the dataset.

Return type `list[int]`

mmpose.apis.webcam.utils.get_mouth_keypoint_ids

`mmpose.apis.webcam.utils.get_mouth_keypoint_ids(model_cfg: mmcv.utils.config.Config) → int`

A helper function to get the mouth keypoint index from the model config.

Parameters `model_cfg (Config)` – pose model config.

Returns The mouth keypoint index

Return type `int`

mmpose.apis.webcam.utils.get_wrist_keypoint_ids

`mmpose.apis.webcam.utils.get_wrist_keypoint_ids(model_cfg: mmcv.utils.config.Config) → Tuple[int, int]`

A helper function to get the keypoint indices of left and right wrists from the model config.

Parameters `model_cfg (Config)` – pose model config.

Returns The keypoint indices of left and right wrists.

Return type `tuple[int, int]`

47.3.3 Event

EventManager

A helper class to manage events.

mmpose.apis.webcam.utils.EventManager

class `mmpose.apis.webcam.utils.EventManager`

A helper class to manage events.

EventManager provides interfaces to register, set, clear and check events by name.

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__()</code>	Initialize self.
<code>clear([event_name, is_keyboard])</code>	Reset the internal flag of an event to False.
<code>is_set([event_name, is_keyboard])</code>	Check whether the internal flag of an event is True.
<code>register_event(event_name[, is_keyboard])</code>	Register an event.
<code>set(event_name[, is_keyboard])</code>	Set the internal flag of an event to True.
<code>wait([event_name, is_keyboard, timeout])</code>	Block until the internal flag of an event is True.
<code>wait_and_handle([event_name, is_keyboard])</code>	Context manager that blocks until an event is set True and then goes into the context.

47.3.4 Misc

<code>copy_and_paste</code>	Copy the image region and paste to the background.
<code>screen_matting</code>	Get screen matting mask.
<code>expand_and_clamp</code>	Expand the bbox and clip it to fit the image shape.
<code>limit_max_fps</code>	A context manager to limit maximum frequency of entering the context.
<code>is_image_file</code>	Check if a path is an image file by its extension.
<code>get_cached_file_path</code>	Loads the Torch serialized object at the given URL.
<code>load_image_from_disk_or_url</code>	Load an image file, from disk or url.
<code>get_config_path</code>	Get config path from an OpenMMLab codebase.

`mmpose.apis.webcam.utils.copy_and_paste`

`mmpose.apis.webcam.utils.copy_and_paste`(*img: numpy.ndarray, background_img: numpy.ndarray, mask: numpy.ndarray, bbox: Optional[List] = None, effect_region: Tuple = (0.2, 0.2, 0.8, 0.8), min_size: Tuple = (20, 20)*) → `numpy.ndarray`

Copy the image region and paste to the background.

Parameters

- **img** (`np.ndarray`) – Image data.
- **background_img** (`np.ndarray`) – Background image data.
- **mask** (`ndarray`) – instance segmentation result.
- **bbox** (`list`, *optional*) – instance bbox in (x1, y1, x2, y2). If not given, the bbox will be obtained by `_find_bbox()`. Default: `None`
- **effect_region** (`tuple`) – The region to apply mask, the coordinates are normalized (x1, y1, x2, y2). Default: (0.2, 0.2, 0.8, 0.8)
- **min_size** (`tuple`) – The minimum region size (w, h) in pixels. Default: (20, 20)

Returns The background with pasted image region.

Return type `np.ndarray`

`mmpose.apis.webcam.utils.screen_matting`

`mmpose.apis.webcam.utils.screen_matting`(*img: numpy.ndarray, color_low: Optional[Tuple] = None, color_high: Optional[Tuple] = None, color: Optional[str] = None*) → `numpy.ndarray`

Get screen matting mask.

Parameters

- **img** (*np.ndarray*) – Image data.
- **color_low** (*tuple*) – Lower limit (b, g, r).
- **color_high** (*tuple*) – Higher limit (b, g, r).
- **color** (*str*) – Support colors include:
 - 'green' or 'g'
 - 'blue' or 'b'
 - 'black' or 'k'
 - 'white' or 'w'

Returns A mask with the same shape of the input image. The value is 0 at the pixels in the matting color range, and 1 everywhere else.

Return type `np.ndarray`

`mmpose.apis.webcam.utils.expand_and_clamp`

`mmpose.apis.webcam.utils.expand_and_clamp`(*box: List, im_shape: Tuple, scale: float = 1.25*) → `List`
Expand the bbox and clip it to fit the image shape.

Parameters

- **box** (*list*) – x1, y1, x2, y2
- **im_shape** (*tuple*) – image shape (h, w, c)
- **scale** (*float*) – expand ratio

Returns x1, y1, x2, y2

Return type `list`

`mmpose.apis.webcam.utils.limit_max_fps`

`mmpose.apis.webcam.utils.limit_max_fps`(*fps: float*)
A context manager to limit maximum frequency of entering the context.

Parameters **fps** (*float*) – The maximum frequency of entering the context

Example::

```
>>> from mmpose.apis.webcam.utils import limit_max_fps
>>> import cv2
```

```
>>> while True:
...     with limit_max_fps(20):
...         cv2.imshow(img) # display image at most 20 fps
```

`mmpose.apis.webcam.utils.is_image_file`

`mmpose.apis.webcam.utils.is_image_file(path: str) → bool`

Check if a path is an image file by its extension.

Parameters `path` (*str*) – The image path.

Returns Whether the path is an image file.

Return type bool

`mmpose.apis.webcam.utils.get_cached_file_path`

`mmpose.apis.webcam.utils.get_cached_file_path(url: str, save_dir: Optional[str] = None, progress: bool = True, check_hash: bool = False, file_name: Optional[str] = None) → str`

Loads the Torch serialized object at the given URL.

If downloaded file is a zip file, it will be automatically decompressed

If the object is already present in `model_dir`, it's deserialized and returned. The default value of `model_dir` is `<hub_dir>/checkpoints` where `hub_dir` is the directory returned by `get_dir()`.

Parameters

- **url** (*str*) – URL of the object to download
- **save_dir** (*str*, *optional*) – directory in which to save the object
- **progress** (*bool*) – whether or not to display a progress bar to stderr. Default: True
- **check_hash** (*bool*) – If True, the filename part of the URL should follow the naming convention `filename-<sha256>.ext` where `<sha256>` is the first eight or more digits of the SHA256 hash of the contents of the file. The hash is used to ensure unique names and to verify the contents of the file. Default: False
- **file_name** (*str*, *optional*) – name for the downloaded file. Filename from `url` will be used if not set. Default: None.

Returns The path to the cached file.

Return type str

`mmpose.apis.webcam.utils.load_image_from_disk_or_url`

`mmpose.apis.webcam.utils.load_image_from_disk_or_url(filename: str, readFlag: int = 1) → numpy.ndarray`

Load an image file, from disk or url.

Parameters

- **filename** (*str*) – file name on the disk or url link
- **readFlag** (*int*) – readFlag for imdecode. Default: `cv2.IMREAD_COLOR`

Returns A loaded image

Return type np.ndarray

mmpose.apis.webcam.utils.get_config_path

`mmpose.apis.webcam.utils.get_config_path(path: str, module_name: str)`

Get config path from an OpenMMLab codebase.

If the path is an existing file, it will be directly returned. If the file doesn't exist, it will be searched in the 'configs' folder of the specified module.

Parameters

- **path** (*str*) – the path of the config file
- **module_name** (*str*) – The module name of an OpenMMLab codebase

Returns The config file path.

Return type str

Example::

```
>>> path = 'configs/_base_/filters/one_euro.py'
>>> get_config_path(path, 'mmpose')
'/home/mmpose/configs/_base_/filters/one_euro.py'
```

MMPOSE.CORE

48.1 evaluation

```
class mmpose.core.evaluation.DistEvalHook(dataloader, start=None, interval=1, by_epoch=True,
                                          save_best=None, rule=None, test_fn=None,
                                          greater_keys=['acc', 'ap', 'ar', 'pck', 'auc', '3dpck', 'p-3dpck',
                                          '3dauc', 'p-3dauc', 'pcp'], less_keys=['loss', 'epe', 'nme',
                                          'mpjpe', 'p-mpjpe', 'n-mpjpe'], broadcast_bn_buffer=True,
                                          tmpdir=None, gpu_collect=False, **eval_kwargs)

class mmpose.core.evaluation.EvalHook(dataloader, start=None, interval=1, by_epoch=True,
                                       save_best=None, rule=None, test_fn=None, greater_keys=['acc',
                                       'ap', 'ar', 'pck', 'auc', '3dpck', 'p-3dpck', '3dauc', 'p-3dauc', 'pcp'],
                                       less_keys=['loss', 'epe', 'nme', 'mpjpe', 'p-mpjpe', 'n-mpjpe'],
                                       **eval_kwargs)

mmpose.core.evaluation.aggregate_scale(feature_maps_list, align_corners=False, project2image=True,
                                       size_projected=None, aggregate_scale='average')
```

Aggregate multi-scale outputs.

Note: batch size: N keypoints num : K heatmap width: W heatmap height: H

Parameters

- **feature_maps_list** (*list*[*Tensor*]) – Aggregated feature maps.
- **project2image** (*bool*) – Option to resize to base scale.
- **size_projected** (*list*[*int*, *int*]) – Base size of heatmaps [w, h].
- **align_corners** (*bool*) – Align corners when performing interpolation.
- **aggregate_scale** (*str*) – Methods to aggregate multi-scale feature maps. Options: ‘average’, ‘unsqueeze_concat’.
 - ‘average’: Get the average of the feature maps.
 - ‘unsqueeze_concat’: **Concatenate the feature maps along new axis.** Default: ‘average’.

Returns Aggregated feature maps.

Return type Tensor

```
mmpose.core.evaluation.aggregate_stage_flip(feature_maps, feature_maps_flip, index=-1,
                                             project2image=True, size_projected=None,
                                             align_corners=False, aggregate_stage='concat',
                                             aggregate_flip='average')
```

Inference the model to get multi-stage outputs (heatmaps & tags), and resize them to base sizes.

Parameters

- **feature_maps** (*list[Tensor]*) – feature_maps can be heatmaps, tags, and pafs.
- **feature_maps_flip** (*list[Tensor] | None*) – flipped feature_maps. feature maps can be heatmaps, tags, and pafs.
- **project2image** (*bool*) – Option to resize to base scale.
- **size_projected** (*list[int, int]*) – Base size of heatmaps [w, h].
- **align_corners** (*bool*) – Align corners when performing interpolation.
- **aggregate_stage** (*str*) – Methods to aggregate multi-stage feature maps. Options: 'concat', 'average'. Default: 'concat'.
 - 'concat': Concatenate the original and the flipped feature maps.
 - 'average': **Get the average of the original and the flipped** feature maps.
- **aggregate_flip** (*str*) – Methods to aggregate the original and the flipped feature maps. Options: 'concat', 'average', 'none'. Default: 'average'.
 - 'concat': Concatenate the original and the flipped feature maps.
 - 'average': **Get the average of the original and the flipped** feature maps..
 - 'none': no flipped feature maps.

Returns Aggregated feature maps with shape [NxKxWxH].

Return type *list[Tensor]*

```
mmpose.core.evaluation.compute_similarity_transform(source_points, target_points)
```

Computes a similarity transform (sR, t) that takes a set of 3D points *source_points* (N x 3) closest to a set of 3D points *target_points*, where R is an 3x3 rotation matrix, t 3x1 translation, s scale. And return the transformed 3D points *source_points_hat* (N x 3). i.e. solves the orthogonal Procrustes problem.

Note: Points number: N

Parameters

- **source_points** (*np.ndarray*) – Source point set with shape [N, 3].
- **target_points** (*np.ndarray*) – Target point set with shape [N, 3].

Returns Transformed source point set with shape [N, 3].

Return type *np.ndarray*

```
mmpose.core.evaluation.flip_feature_maps(feature_maps, flip_index=None)
```

Flip the feature maps and swap the channels.

Parameters

- **feature_maps** (*list[Tensor]*) – Feature maps.
- **flip_index** (*list[int] | None*) – Channel-flip indexes. If None, do not flip channels.

Returns Flipped feature_maps.

Return type list[`Tensor`]

`mmpose.core.evaluation.get_group_preds(grouped_joints, center, scale, heatmap_size, use_udp=False)`
Transform the grouped joints back to the image.

Parameters

- **grouped_joints** (`list`) – Grouped person joints.
- **center** (`np.ndarray[2,]`) – Center of the bounding box (x, y).
- **scale** (`np.ndarray[2,]`) – Scale of the bounding box wrt [width, height].
- **heatmap_size** (`np.ndarray[2,]`) – Size of the destination heatmaps.
- **use_udp** (`bool`) – Unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR'2020).

Returns List of the pose result for each person.

Return type list

`mmpose.core.evaluation.keypoint_3d_auc(pred, gt, mask, alignment='none')`
Calculate the Area Under the Curve (3DAUC) computed for a range of 3DPCK thresholds.

Paper ref: [Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision' 3DV'2017](#).
This implementation is derived from `mpii_compute_3d_pck.m`, which is provided as part of the MPI-INF-3DHP test data release.

Note: batch_size: N num_keypoints: K keypoint_dims: C

Parameters

- **pred** (`np.ndarray[N, K, C]`) – Predicted keypoint location.
- **gt** (`np.ndarray[N, K, C]`) – Groundtruth keypoint location.
- **mask** (`np.ndarray[N, K]`) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **alignment** (`str, optional`) – method to align the prediction with the groundtruth. Supported options are:
 - 'none': no alignment will be applied
 - 'scale': align in the least-square sense in scale
 - 'procrustes': align in the least-square sense in scale, rotation and translation.

Returns AUC computed for a range of 3DPCK thresholds.

Return type auc

`mmpose.core.evaluation.keypoint_3d_pck(pred, gt, mask, alignment='none', threshold=0.15)`
Calculate the Percentage of Correct Keypoints (3DPCK) w. or w/o rigid alignment.

Paper ref: [Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision' 3DV'2017](#). .

Note:

- batch_size: N

- `num_keypoints`: K
 - `keypoint_dims`: C
-

Parameters

- **pred** (*np.ndarray*[N, K, C]) – Predicted keypoint location.
- **gt** (*np.ndarray*[N, K, C]) – Groundtruth keypoint location.
- **mask** (*np.ndarray*[N, K]) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **alignment** (*str, optional*) – method to align the prediction with the groundtruth. Supported options are:
 - 'none': no alignment will be applied
 - 'scale': align in the least-square sense in scale
 - '**procrustes**': align in the least-square sense in scale, rotation and translation.
- **threshold** – If L2 distance between the prediction and the groundtruth is less than threshold, the predicted result is considered as correct. Default: 0.15 (m).

Returns percentage of correct keypoints.

Return type pck

`mmpose.core.evaluation.keypoint_auc(pred, gt, mask, normalize, num_step=20)`

Calculate the pose accuracy of PCK for each individual keypoint and the averaged accuracy across all keypoints for coordinates.

Note:

- `batch_size`: N
 - `num_keypoints`: K
-

Parameters

- **pred** (*np.ndarray*[N, K, 2]) – Predicted keypoint location.
- **gt** (*np.ndarray*[N, K, 2]) – Groundtruth keypoint location.
- **mask** (*np.ndarray*[N, K]) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **normalize** (*float*) – Normalization factor.

Returns Area under curve.

Return type float

`mmpose.core.evaluation.keypoint_epe(pred, gt, mask)`

Calculate the end-point error.

Note:

- `batch_size`: N

- `num_keypoints`: K

Parameters

- **pred** (`np.ndarray[N, K, 2]`) – Predicted keypoint location.
- **gt** (`np.ndarray[N, K, 2]`) – Groundtruth keypoint location.
- **mask** (`np.ndarray[N, K]`) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.

Returns Average end-point error.

Return type float

`mmpose.core.evaluation.keypoint_mpjpe(pred, gt, mask, alignment='none')`

Calculate the mean per-joint position error (MPJPE) and the error after rigid alignment with the ground truth (P-MPJPE).

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `keypoint_dims`: C
-

Parameters

- **pred** (`np.ndarray`) – Predicted keypoint location with shape [N, K, C].
- **gt** (`np.ndarray`) – Groundtruth keypoint location with shape [N, K, C].
- **mask** (`np.ndarray`) – Visibility of the target with shape [N, K]. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **alignment** (`str`, *optional*) – method to align the prediction with the groundtruth. Supported options are:
 - `'none'`: no alignment will be applied
 - `'scale'`: align in the least-square sense in scale
 - **`'procrustes'`: align in the least-square sense in** scale, rotation and translation.

Returns

A tuple containing joint position errors

- (`float` | `np.ndarray`): mean per-joint position error (mpjpe).
- (`float` | `np.ndarray`): **mpjpe after rigid alignment with the** ground truth (p-mpjpe).

Return type tuple

`mmpose.core.evaluation.keypoint_pck_accuracy(pred, gt, mask, thr, normalize)`

Calculate the pose accuracy of PCK for each individual keypoint and the averaged accuracy across all keypoints for coordinates.

Note: PCK metric measures accuracy of the localization of the body joints. The distances between predicted positions and the ground-truth ones are typically normalized by the bounding box size. The threshold (thr) of the normalized distance is commonly set as 0.05, 0.1 or 0.2 etc.

- `batch_size`: N
 - `num_keypoints`: K
-

Parameters

- **pred** (`np.ndarray[N, K, 2]`) – Predicted keypoint location.
- **gt** (`np.ndarray[N, K, 2]`) – Groundtruth keypoint location.
- **mask** (`np.ndarray[N, K]`) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **thr** (`float`) – Threshold of PCK calculation.
- **normalize** (`np.ndarray[N, 2]`) – Normalization factor for H&W.

Returns

A tuple containing keypoint accuracy.

- `acc` (`np.ndarray[K]`): Accuracy of each keypoint.
- `avg_acc` (`float`): Averaged accuracy across all keypoints.
- `cnt` (`int`): Number of valid keypoints.

Return type tuple

```
mmpose.core.evaluation.keypoints_from_heatmaps(heatmaps, center, scale, unbiased=False,
                                              post_process='default', kernel=11,
                                              valid_radius_factor=0.0546875, use_udp=False,
                                              target_type='GaussianHeatmap')
```

Get final keypoint predictions from heatmaps and transform them back to the image.

Note:

- batch size: N
 - num keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **heatmaps** (`np.ndarray[N, K, H, W]`) – model predicted heatmaps.
- **center** (`np.ndarray[N, 2]`) – Center of the bounding box (x, y).
- **scale** (`np.ndarray[N, 2]`) – Scale of the bounding box wrt height/width.
- **post_process** (`str/None`) – Choice of methods to post-process heatmaps. Currently supported: None, 'default', 'unbiased', 'megvii'.

- **unbiased** (*bool*) – Option to use unbiased decoding. Mutually exclusive with `megvii`. Note: this arg is deprecated and `unbiased=True` can be replaced by `post_process='unbiased'` Paper ref: Zhang et al. Distribution-Aware Coordinate Representation for Human Pose Estimation (CVPR 2020).
- **kernel** (*int*) – Gaussian kernel size (K) for modulation, which should match the heatmap gaussian sigma when training. K=17 for sigma=3 and k=11 for sigma=2.
- **valid_radius_factor** (*float*) – The radius factor of the positive area in classification heatmap for UDP.
- **use_udp** (*bool*) – Use unbiased data processing.
- **target_type** (*str*) – ‘GaussianHeatmap’ or ‘CombinedTarget’. GaussianHeatmap: Classification target with gaussian distribution. CombinedTarget: The combination of classification target (response map) and regression target (offset map). Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

Returns

A tuple containing keypoint predictions and scores.

- `preds` (`np.ndarray[N, K, 2]`): Predicted keypoint location in images.
- `maxvals` (`np.ndarray[N, K, 1]`): Scores (confidence) of the keypoints.

Return type tuple

`mmpose.core.evaluation.keypoints_from_heatmaps3d(heatmaps, center, scale)`

Get final keypoint predictions from 3d heatmaps and transform them back to the image.

Note:

- batch size: N
 - num keypoints: K
 - heatmap depth size: D
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **heatmaps** (`np.ndarray[N, K, D, H, W]`) – model predicted heatmaps.
- **center** (`np.ndarray[N, 2]`) – Center of the bounding box (x, y).
- **scale** (`np.ndarray[N, 2]`) – Scale of the bounding box wrt height/width.

Returns

A tuple containing keypoint predictions and scores.

- `preds` (`np.ndarray[N, K, 3]`): Predicted 3d keypoint location in images.
- `maxvals` (`np.ndarray[N, K, 1]`): Scores (confidence) of the keypoints.

Return type tuple

`mmpose.core.evaluation.keypoints_from_regression(regression_preds, center, scale, img_size)`
Get final keypoint predictions from regression vectors and transform them back to the image.

Note:

- `batch_size`: N
 - `num_keypoints`: K
-

Parameters

- **`regression_preds`** (`np.ndarray[N, K, 2]`) – model prediction.
- **`center`** (`np.ndarray[N, 2]`) – Center of the bounding box (x, y).
- **`scale`** (`np.ndarray[N, 2]`) – Scale of the bounding box wrt height/width.
- **`img_size`** (`list(img_width, img_height)`) – model input image size.

Returns

- `preds` (`np.ndarray[N, K, 2]`): Predicted keypoint location in images.
- `maxvals` (`np.ndarray[N, K, 1]`): Scores (confidence) of the keypoints.

Return type tuple

`mmpose.core.evaluation.multilabel_classification_accuracy(pred, gt, mask, thr=0.5)`
Get multi-label classification accuracy.

Note:

- batch size: N
 - label number: L
-

Parameters

- **`pred`** (`np.ndarray[N, L, 2]`) – model predicted labels.
- **`gt`** (`np.ndarray[N, L, 2]`) – ground-truth labels.
- **`mask`** (`np.ndarray[N, 1]` or `np.ndarray[N, L]`) – reliability of
- **`labels`**. (*ground-truth*) –

Returns multi-label classification accuracy.

Return type float

`mmpose.core.evaluation.pose_pck_accuracy(output, target, mask, thr=0.05, normalize=None)`
Calculate the pose accuracy of PCK for each individual keypoint and the averaged accuracy across all keypoints from heatmaps.

Note: PCK metric measures accuracy of the localization of the body joints. The distances between predicted positions and the ground-truth ones are typically normalized by the bounding box size. The threshold (`thr`) of the normalized distance is commonly set as 0.05, 0.1 or 0.2 etc.

- `batch_size`: N

- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **output** (`np.ndarray[N, K, H, W]`) – Model output heatmaps.
- **target** (`np.ndarray[N, K, H, W]`) – Groundtruth heatmaps.
- **mask** (`np.ndarray[N, K]`) – Visibility of the target. False for invisible joints, and True for visible. Invisible joints will be ignored for accuracy calculation.
- **thr** (`float`) – Threshold of PCK calculation. Default 0.05.
- **normalize** (`np.ndarray[N, 2]`) – Normalization factor for H&W.

Returns

A tuple containing keypoint accuracy.

- `np.ndarray[K]`: Accuracy of each keypoint.
- `float`: Averaged accuracy across all keypoints.
- `int`: Number of valid keypoints.

Return type tuple

`mmpose.core.evaluation.post_dark_udp(coords, batch_heatmaps, kernel=3)`

DARK post-processing. Implemented by udp. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020). Zhang et al. Distribution-Aware Coordinate Representation for Human Pose Estimation (CVPR 2020).

Note:

- batch size: B
- num keypoints: K
- num persons: N
- height of heatmaps: H
- width of heatmaps: W

B=1 for bottom_up paradigm where all persons share the same heatmap. B=N for top_down paradigm where each person has its own heatmaps.

Parameters

- **coords** (`np.ndarray[N, K, 2]`) – Initial coordinates of human pose.
- **batch_heatmaps** (`np.ndarray[B, K, H, W]`) – batch_heatmaps
- **kernel** (`int`) – Gaussian kernel size (K) for modulation.

Returns Refined coordinates.

Return type `np.ndarray([N, K, 2])`

`mmpose.core.evaluation.split_ae_outputs(outputs, num_joints, with_heatmaps, with_ae, select_output_index)`

Split multi-stage outputs into heatmaps & tags.

Parameters

- **outputs** (*list(Tensor)*) – Outputs of network
- **num_joints** (*int*) – Number of joints
- **with_heatmaps** (*list[bool]*) – Option to output heatmaps for different stages.
- **with_ae** (*list[bool]*) – Option to output ae tags for different stages.
- **select_output_index** (*list[int]*) – Output keep the selected index

Returns

A tuple containing multi-stage outputs.

- *list[Tensor]*: multi-stage heatmaps.
- *list[Tensor]*: multi-stage tags.

Return type tuple

48.2 fp16

`class mmpose.core.fp16.Fp16OptimizerHook(grad_clip=None, coalesce=True, bucket_size_mb=-1, loss_scale=512.0, distributed=True)`

FP16 optimizer hook.

The steps of fp16 optimizer is as follows. 1. Scale the loss value. 2. BP in the fp16 model. 2. Copy gradients from fp16 model to fp32 weights. 3. Update fp32 weights. 4. Copy updated parameters from fp32 weights to fp16 model.

Refer to <https://arxiv.org/abs/1710.03740> for more details.

Parameters **loss_scale** (*float*) – Scale factor multiplied with loss.

after_train_iter(*runner*)

Backward optimization steps for Mixed Precision Training.

1. Scale the loss by a scale factor.
2. Backward the loss to obtain the gradients (fp16).
3. Copy gradients from the model to the fp32 weight copy.
4. Scale the gradients back and update the fp32 weight copy.
5. Copy back the params from fp32 weight copy to the fp16 model.

Parameters **runner** (*mmpcv.Runner*) – The underlines training runner.

before_run(*runner*)

Preparing steps before Mixed Precision Training.

1. Make a master copy of fp32 weights for optimization.
2. Convert the main model from fp32 to fp16.

Parameters **runner** (*mmpcv.Runner*) – The underlines training runner.

static copy_grads_to_fp32(*fp16_net, fp32_weights*)

Copy gradients from fp16 model to fp32 weight copy.

static copy_params_to_fp16(*fp16_net, fp32_weights*)

Copy updated params from fp32 weight copy to fp16 model.

mmpose.core.fp16.auto_fp16(*apply_to=None, out_fp32=False*)

Decorator to enable fp16 training automatically.

This decorator is useful when you write custom modules and want to support mixed precision training. If inputs arguments are fp32 tensors, they will be converted to fp16 automatically. Arguments other than fp32 tensors are ignored.

Parameters

- **apply_to** (*Iterable, optional*) – The argument names to be converted. *None* indicates all arguments.
- **out_fp32** (*bool*) – Whether to convert the output back to fp32.

Example

```
>>> import torch.nn as nn
>>> class MyModule1(nn.Module):
>>>
>>>     # Convert x and y to fp16
>>>     @auto_fp16()
>>>     def forward(self, x, y):
>>>         pass
```

```
>>> import torch.nn as nn
>>> class MyModule2(nn.Module):
>>>
>>>     # convert pred to fp16
>>>     @auto_fp16(apply_to=('pred', ))
>>>     def do_something(self, pred, others):
>>>         pass
```

mmpose.core.fp16.cast_tensor_type(*inputs, src_type, dst_type*)

Recursively convert Tensor in inputs from *src_type* to *dst_type*.

Parameters

- **inputs** – Inputs that to be casted.
- **src_type** (*torch.dtype*) – Source type.
- **dst_type** (*torch.dtype*) – Destination type.

Returns The same type with inputs, but all contained Tensors have been cast.

mmpose.core.fp16.force_fp32(*apply_to=None, out_fp16=False*)

Decorator to convert input arguments to fp32 in force.

This decorator is useful when you write custom modules and want to support mixed precision training. If there are some inputs that must be processed in fp32 mode, then this decorator can handle it. If inputs arguments are fp16 tensors, they will be converted to fp32 automatically. Arguments other than fp16 tensors are ignored.

Parameters

- **apply_to** (*Iterable*, *optional*) – The argument names to be converted. *None* indicates all arguments.
- **out_fp16** (*bool*) – Whether to convert the output back to fp16.

Example

```
>>> import torch.nn as nn
>>> class MyModule1(nn.Module):
>>>
>>>     # Convert x and y to fp32
>>>     @force_fp32()
>>>     def loss(self, x, y):
>>>         pass
```

```
>>> import torch.nn as nn
>>> class MyModule2(nn.Module):
>>>
>>>     # convert pred to fp32
>>>     @force_fp32(apply_to=('pred', ))
>>>     def post_process(self, pred, others):
>>>         pass
```

`mmpose.core.fp16.wrap_fp16_model(model)`

Wrap the FP32 model to FP16.

1. Convert FP32 model to FP16.
2. Remain some necessary layers to be FP32, e.g., normalization layers.

Parameters `model` (*nn.Module*) – Model in FP32.

48.3 utils

class `mmpose.core.utils.ModelSetEpochHook`

The hook that tells model the current epoch in training.

class `mmpose.core.utils.WeightNormClipHook(max_norm=1.0, module_param_names='weight')`

Apply weight norm clip regularization.

The module's parameter will be clip to a given maximum norm before each forward pass.

Parameters

- **max_norm** (*float*) – The maximum norm of the parameter.
- **module_param_names** (*str/list*) – The parameter name (or name list) to apply weight norm clip.

hook (*module*, *_input*)

Hook function.

property `hook_type`

Hook type Subclasses should overwrite this function to return a string value in.

{*forward*, *forward_pre*, *backward*}

`mmpose.core.utils.allreduce_grads(params, coalesce=True, bucket_size_mb=-1)`
Allreduce gradients.

Parameters

- **params** (`list[torch.Parameters]`) – List of parameters of a model
- **coalesce** (`bool, optional`) – Whether allreduce parameters as a whole. Default: True.
- **bucket_size_mb** (`int, optional`) – Size of bucket, the unit is MB. Default: -1.

`mmpose.core.utils.sync_random_seed(seed=None, device='cuda')`
Make sure different ranks share the same seed.

All workers must call this function, otherwise it will deadlock. This method is generally used in *DistributedSampler*, because the seed should be identical across all processes in the distributed group. In distributed sampling, different ranks should sample non-overlapped data in the dataset. Therefore, this function is used to make sure that each rank shuffles the data indices in the same order based on the same seed. Then different ranks could use different indices to select non-overlapped data from the same data list. :param seed: The seed. Default to None. :type seed: int, Optional :param device: The device where the seed will be put on.

Default to 'cuda'.

Returns Seed to be used.

Return type int

48.4 post_processing

`class mmpose.core.post_processing.Smoother(filter_cfg: Union[Dict, str], keypoint_dim: int = 2, keypoint_key: str = 'keypoints')`

Smoother to apply temporal smoothing on pose estimation results with a filter.

Note: T: The temporal length of the pose sequence K: The keypoint number of each target C: The keypoint coordinate dimension

Parameters

- **filter_cfg** (`dict | str`) – The filter config. See example config files in *configs/_base_/filters/* for details. Alternatively a config file path can be accepted and the config will be loaded.
- **keypoint_dim** (`int`) – The keypoint coordinate dimension, which is also indicated as C. Default: 2
- **keypoint_key** (`str`) – The dict key of the keypoints in the pose results. Default: 'keypoints'

Example

```

>>> import numpy as np
>>> # Build dummy pose result
>>> results = []
>>> for t in range(10):
>>>     results_t = []
>>>     for track_id in range(2):
>>>         result = {
>>>             'track_id': track_id,
>>>             'keypoints': np.random.rand(17, 3)
>>>         }
>>>         results_t.append(result)
>>>     results.append(results_t)
>>> # Example 1: Smooth multi-frame pose results offline.
>>> filter_cfg = dict(type='GaussianFilter', window_size=3)
>>> smoother = Smoother(filter_cfg, keypoint_dim=2)
>>> smoothed_results = smoother.smooth(results)
>>> # Example 2: Smooth pose results online frame-by-frame
>>> filter_cfg = dict(type='GaussianFilter', window_size=3)
>>> smoother = Smoother(filter_cfg, keypoint_dim=2)
>>> for result_t in results:
>>>     smoothed_result_t = smoother.smooth(result_t)

```

smooth(results)

Apply temporal smoothing on pose estimation sequences.

Parameters **results** (*list[dict] | list[list[dict]]*) – The pose results of a single frame (non-nested list) or multiple frames (nested list). The result of each target is a dict, which should contains:

- **track_id** (optional, Any): The track ID of the target
- **keypoints** (*np.ndarray*): The keypoint coordinates in [K, C]

Returns Temporal smoothed pose results, which has the same data structure as the input's.

Return type (*list[dict] | list[list[dict]]*)

`mmpose.core.post_processing.affine_transform(pt, trans_mat)`

Apply an affine transformation to the points.

Parameters

- **pt** (*np.ndarray*) – a 2 dimensional point to be transformed
- **trans_mat** (*np.ndarray*) – 2x3 matrix of an affine transform

Returns Transformed points.

Return type *np.ndarray*

`mmpose.core.post_processing.flip_back(output_flipped, flip_pairs, target_type='GaussianHeatmap')`

Flip the flipped heatmaps back to the original form.

Note:

- **batch_size**: N
- **num_keypoints**: K

- heatmap height: H
- heatmap width: W

Parameters

- **output_flipped** (*np.ndarray*[N, K, H, W]) – The output heatmaps obtained from the flipped images.
- **flip_pairs** (*list[tuple()]*) – Pairs of keypoints which are mirrored (for example, left ear – right ear).
- **target_type** (*str*) – GaussianHeatmap or CombinedTarget

Returns heatmaps that flipped back to the original image

Return type *np.ndarray*

`mmpose.core.post_processing.fliplr_joints(joints_3d, joints_3d_visible, img_width, flip_pairs)`
 Flip human joints horizontally.

Note:

- num_keypoints: K
-

Parameters

- **joints_3d** (*np.ndarray* ([K, 3])) – Coordinates of keypoints.
- **joints_3d_visible** (*np.ndarray* ([K, 1])) – Visibility of keypoints.
- **img_width** (*int*) – Image width.
- **flip_pairs** (*list[tuple]*) – Pairs of keypoints which are mirrored (for example, left ear and right ear).

Returns

Flipped human joints.

- **joints_3d_flipped** (*np.ndarray* ([K, 3])): Flipped joints.
- **joints_3d_visible_flipped** (*np.ndarray* ([K, 1])): Joint visibility.

Return type *tuple*

`mmpose.core.post_processing.fliplr_regression(regression, flip_pairs, center_mode='static', center_x=0.5, center_index=0)`

Flip human joints horizontally.

Note:

- batch_size: N
 - num_keypoint: K
-

Parameters

- **regression** (*np.ndarray*([... , K, C])) – Coordinates of keypoints, where K is the joint number and C is the dimension. Example shapes are:
 - [N, K, C]: a batch of keypoints where N is the batch size.
 - [N, T, K, C]: a batch of pose sequences, where T is the frame number.
- **flip_pairs** (*list[tuple()]*) – Pairs of keypoints which are mirrored (for example, left ear – right ear).
- **center_mode** (*str*) – The mode to set the center location on the x-axis to flip around. Options are:
 - static: use a static x value (see center_x also)
 - root: use a root joint (see center_index also)
- **center_x** (*float*) – Set the x-axis location of the flip center. Only used when center_mode=static.
- **center_index** (*int*) – Set the index of the root joint, whose x location will be used as the flip center. Only used when center_mode=root.

Returns Flipped joints.

Return type *np.ndarray*([... , K, C])

`mmpose.core.post_processing.get_affine_transform(center, scale, rot, output_size, shift=(0.0, 0.0), inv=False)`

Get the affine transform matrix, given the center/scale/rot/output_size.

Parameters

- **center** (*np.ndarray*[2,]) – Center of the bounding box (x, y).
- **scale** (*np.ndarray*[2,]) – Scale of the bounding box wrt [width, height].
- **rot** (*float*) – Rotation angle (degree).
- **output_size** (*np.ndarray*[2,] | *list*(2,)) – Size of the destination heatmaps.
- **shift** (0-100%) – Shift translation ratio wrt the width/height. Default (0., 0.).
- **inv** (*bool*) – Option to inverse the affine transform direction. (inv=False: src->dst or inv=True: dst->src)

Returns The transform matrix.

Return type *np.ndarray*

`mmpose.core.post_processing.get_warp_matrix(theta, size_input, size_dst, size_target)`

Calculate the transformation matrix under the constraint of unbiased. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

Parameters

- **theta** (*float*) – Rotation angle in degrees.
- **size_input** (*np.ndarray*) – Size of input image [w, h].
- **size_dst** (*np.ndarray*) – Size of output image [w, h].
- **size_target** (*np.ndarray*) – Size of ROI in input plane [w, h].

Returns A matrix for transformation.

Return type *np.ndarray*

`mmpose.core.post_processing.nearby_joints_nms(kpts_db, dist_thr, num_nearby_joints_thr=None, score_per_joint=False, max_dets=-1)`

Nearby joints NMS implementations.

Parameters

- **kpts_db** (*list[dict]*) – keypoints and scores.
- **dist_thr** (*float*) – threshold for judging whether two joints are close.
- **num_nearby_joints_thr** (*int*) – threshold for judging whether two instances are close.
- **max_dets** (*int*) – max number of detections to keep.
- **score_per_joint** (*bool*) – the input scores (in kpts_db) are per joint scores.

Returns indexes to keep.

Return type np.ndarray

`mmpose.core.post_processing.oks_iou(g, d, a_g, a_d, sigmas=None, vis_thr=None)`

Calculate oks ious.

Parameters

- **g** – Ground truth keypoints.
- **d** – Detected keypoints.
- **a_g** – Area of the ground truth object.
- **a_d** – Area of the detected object.
- **sigmas** – standard deviation of keypoint labelling.
- **vis_thr** – threshold of the keypoint visibility.

Returns The oks ious.

Return type list

`mmpose.core.post_processing.oks_nms(kpts_db, thr, sigmas=None, vis_thr=None, score_per_joint=False)`

OKS NMS implementations.

Parameters

- **kpts_db** – keypoints.
- **thr** – Retain overlap < thr.
- **sigmas** – standard deviation of keypoint labelling.
- **vis_thr** – threshold of the keypoint visibility.
- **score_per_joint** – the input scores (in kpts_db) are per joint scores

Returns indexes to keep.

Return type np.ndarray

`mmpose.core.post_processing.rotate_point(pt, angle_rad)`

Rotate a point by an angle.

Parameters

- **pt** (*list[float]*) – 2 dimensional point to be rotated
- **angle_rad** (*float*) – rotation angle by radian

Returns Rotated point.

Return type list[float]

`mmpose.core.post_processing.soft oks_nms(kpts_db, thr, max_dets=20, sigmas=None, vis_thr=None, score_per_joint=False)`

Soft OKS NMS implementations.

Parameters

- **kpts_db** – keypoints and scores.
- **thr** – retain oks overlap < thr.
- **max_dets** – max number of detections to keep.
- **sigmas** – Keypoint labelling uncertainty.
- **score_per_joint** – the input scores (in kpts_db) are per joint scores

Returns indexes to keep.

Return type np.ndarray

`mmpose.core.post_processing.transform_preds(coords, center, scale, output_size, use_udp=False)`

Get final keypoint predictions from heatmaps and apply scaling and translation to map them back to the image.

Note: num_keypoints: K

Parameters

- **coords** (`np.ndarray[K, ndims]`) –
 - If ndims=2, corrd are predicted keypoint location.
 - If ndims=4, corrd are composed of (x, y, scores, tags)
 - If ndims=5, corrd are composed of (x, y, scores, tags, flipped_tags)
- **center** (`np.ndarray[2,]`) – Center of the bounding box (x, y).
- **scale** (`np.ndarray[2,]`) – Scale of the bounding box wrt [width, height].
- **output_size** (`np.ndarray[2,] | list(2,)`) – Size of the destination heatmaps.
- **use_udp** (`bool`) – Use unbiased data processing

Returns Predicted coordinates in the images.

Return type np.ndarray

`mmpose.core.post_processing.warp_affine_joints(joints, mat)`

Apply affine transformation defined by the transform matrix on the joints.

Parameters

- **joints** (`np.ndarray[... , 2]`) – Origin coordinate of joints.
- **mat** (`np.ndarray[3, 2]`) – The affine matrix.

Returns Result coordinate of joints.

Return type np.ndarray[... , 2]

MMPOSE.MODELS

49.1 backbones

class `mmpose.models.backbones.AlexNet`(*num_classes=-1*)
AlexNet backbone.

The input for AlexNet is a 224x224 RGB image.

Parameters `num_classes` (*int*) – number of classes for classification. The default value is -1, which uses the backbone as a feature extractor without the top classifier.

forward(*x*)
Forward function.

Parameters `x` (*Tensor* | *tuple[Tensor]*) – `x` could be a `torch.Tensor` or a tuple of `torch.Tensor`, containing input data for forward computation.

class `mmpose.models.backbones.CPM`(*in_channels*, *out_channels*, *feat_channels=128*, *middle_channels=32*, *num_stages=6*, *norm_cfg={'requires_grad': True, 'type': 'BN'}*)

CPM backbone.

Convolutional Pose Machines. More details can be found in the [paper](#).

Parameters

- **in_channels** (*int*) – The input channels of the CPM.
- **out_channels** (*int*) – The output channels of the CPM.
- **feat_channels** (*int*) – Feature channel of each CPM stage.
- **middle_channels** (*int*) – Feature channel of conv after the middle stage.
- **num_stages** (*int*) – Number of stages.
- **norm_cfg** (*dict*) – Dictionary to construct and config norm layer.

Example

```
>>> from mmpose.models import CPM
>>> import torch
>>> self = CPM(3, 17)
>>> self.eval()
>>> inputs = torch.rand(1, 3, 368, 368)
>>> level_outputs = self.forward(inputs)
>>> for level_output in level_outputs:
...     print(tuple(level_output.shape))
(1, 17, 46, 46)
(1, 17, 46, 46)
(1, 17, 46, 46)
(1, 17, 46, 46)
(1, 17, 46, 46)
(1, 17, 46, 46)
(1, 17, 46, 46)
```

forward(*x*)

Model forward function.

init_weights(*pretrained=None*)

Initialize the weights in backbone.

Parameters *pretrained* (*str*, *optional*) – Path to pre-trained weights. Defaults to None.

```
class mmpose.models.backbones.HRFormer(extra, in_channels=3, conv_cfg=None,
                                       norm_cfg={'requires_grad': True, 'type': 'BN'},
                                       transformer_norm_cfg={'eps': 1e-06, 'type': 'LN'},
                                       norm_eval=False, with_cp=False, zero_init_residual=False,
                                       frozen_stages=-1)
```

HRFormer backbone.

This backbone is the implementation of [HRFormer: High-Resolution Transformer for Dense Prediction](#).

Parameters

- ***extra* (*dict*)** – Detailed configuration for each stage of HRNet. There must be 4 stages, the configuration for each stage must have 5 keys:
 - *num_modules* (*int*): The number of HRModule in this stage.
 - *num_branches* (*int*): The number of branches in the HRModule.
 - *block* (*str*): The type of block.
 - ***num_blocks* (*tuple*): The number of blocks in each branch.** The length must be equal to *num_branches*.
 - ***num_channels* (*tuple*): The number of channels in each branch.** The length must be equal to *num_branches*.
- ***in_channels* (*int*)** – Number of input image channels. Normally 3.
- ***conv_cfg* (*dict*)** – Dictionary to construct and config conv layer. Default: None.
- ***norm_cfg* (*dict*)** – Config of norm layer. Use *SyncBN* by default.
- ***transformer_norm_cfg* (*dict*)** – Config of transformer norm layer. Use *LN* by default.
- ***norm_eval* (*bool*)** – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.

- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.

Example

```
>>> from mmpose.models import HRFormer
>>> import torch
>>> extra = dict(
>>>     stage1=dict(
>>>         num_modules=1,
>>>         num_branches=1,
>>>         block='BOTTLENECK',
>>>         num_blocks=(2, ),
>>>         num_channels=(64, )),
>>>     stage2=dict(
>>>         num_modules=1,
>>>         num_branches=2,
>>>         block='HRFORMER',
>>>         window_sizes=(7, 7),
>>>         num_heads=(1, 2),
>>>         mlp_ratios=(4, 4),
>>>         num_blocks=(2, 2),
>>>         num_channels=(32, 64)),
>>>     stage3=dict(
>>>         num_modules=4,
>>>         num_branches=3,
>>>         block='HRFORMER',
>>>         window_sizes=(7, 7, 7),
>>>         num_heads=(1, 2, 4),
>>>         mlp_ratios=(4, 4, 4),
>>>         num_blocks=(2, 2, 2),
>>>         num_channels=(32, 64, 128)),
>>>     stage4=dict(
>>>         num_modules=2,
>>>         num_branches=4,
>>>         block='HRFORMER',
>>>         window_sizes=(7, 7, 7, 7),
>>>         num_heads=(1, 2, 4, 8),
>>>         mlp_ratios=(4, 4, 4, 4),
>>>         num_blocks=(2, 2, 2, 2),
>>>         num_channels=(32, 64, 128, 256)))
>>> self = HRFormer(extra, in_channels=1)
>>> self.eval()
>>> inputs = torch.rand(1, 1, 32, 32)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 32, 8, 8)
(1, 64, 4, 4)
(1, 128, 2, 2)
```

(continues on next page)

(continued from previous page)

(1, 256, 1, 1)

```
class mmpose.models.backbones.HRNet(extra, in_channels=3, conv_cfg=None, norm_cfg={'type': 'BN'},
                                     norm_eval=False, with_cp=False, zero_init_residual=False,
                                     frozen_stages=-1)
```

HRNet backbone.

High-Resolution Representations for Labeling Pixels and Regions

Parameters

- **extra** (*dict*) – detailed configuration for each stage of HRNet.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **conv_cfg** (*dict*) – dictionary to construct and config conv layer.
- **norm_cfg** (*dict*) – dictionary to construct and config norm layer.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed.
- **zero_init_residual** (*bool*) – whether to use zero init for last norm layer in resblocks to let them behave as identity.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.

Example

```
>>> from mmpose.models import HRNet
>>> import torch
>>> extra = dict(
>>>     stage1=dict(
>>>         num_modules=1,
>>>         num_branches=1,
>>>         block='BOTTLENECK',
>>>         num_blocks=(4, ),
>>>         num_channels=(64, )),
>>>     stage2=dict(
>>>         num_modules=1,
>>>         num_branches=2,
>>>         block='BASIC',
>>>         num_blocks=(4, 4),
>>>         num_channels=(32, 64)),
>>>     stage3=dict(
>>>         num_modules=4,
>>>         num_branches=3,
>>>         block='BASIC',
>>>         num_blocks=(4, 4, 4),
>>>         num_channels=(32, 64, 128)),
>>>     stage4=dict(
>>>         num_modules=3,
```

(continues on next page)

(continued from previous page)

```

>>>         num_branches=4,
>>>         block='BASIC',
>>>         num_blocks=(4, 4, 4, 4),
>>>         num_channels=(32, 64, 128, 256)))
>>> self = HRNet(extra, in_channels=1)
>>> self.eval()
>>> inputs = torch.rand(1, 1, 32, 32)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 32, 8, 8)

```

forward(x)

Forward function.

init_weights(pretrained=None)

Initialize the weights in backbone.

Parameters **pretrained** (*str*, *optional*) – Path to pre-trained weights. Defaults to None.**property norm1**

the normalization layer named “norm1”

Type nn.Module**property norm2**

the normalization layer named “norm2”

Type nn.Module**train(mode=True)**

Convert the model into training mode.

```

class mmpose.models.backbones.HourglassAENet(downsample_times=4, num_stacks=1, out_channels=34,
                                              stage_channels=(256, 384, 512, 640, 768),
                                              feat_channels=256, norm_cfg={'requires_grad': True,
                                                                              'type': 'BN'})

```

Hourglass-AE Network proposed by Newell et al.

Associative Embedding: End-to-End Learning for Joint Detection and Grouping.

More details can be found in the [paper](#) .**Parameters**

- **downsample_times** (*int*) – Downsample times in a HourglassModule.
- **num_stacks** (*int*) – Number of HourglassModule modules stacked, 1 for Hourglass-52, 2 for Hourglass-104.
- **stage_channels** (*list[int]*) – Feature channel of each sub-module in a HourglassModule.
- **stage_blocks** (*list[int]*) – Number of sub-modules stacked in a HourglassModule.
- **feat_channels** (*int*) – Feature channel of conv after a HourglassModule.
- **norm_cfg** (*dict*) – Dictionary to construct and config norm layer.

Example

```
>>> from mmpose.models import HourglassAENet
>>> import torch
>>> self = HourglassAENet()
>>> self.eval()
>>> inputs = torch.rand(1, 3, 512, 512)
>>> level_outputs = self.forward(inputs)
>>> for level_output in level_outputs:
...     print(tuple(level_output.shape))
(1, 34, 128, 128)
```

forward(x)

Model forward function.

init_weights(pretrained=None)

Initialize the weights in backbone.

Parameters *pretrained* (str, optional) – Path to pre-trained weights. Defaults to None.

```
class mmpose.models.backbones.HourglassNet(downsample_times=5, num_stacks=2, stage_channels=(256,
256, 384, 384, 384, 512), stage_blocks=(2, 2, 2, 2, 2, 4),
feat_channel=256, norm_cfg={'requires_grad': True, 'type':
'BN'})
```

HourglassNet backbone.

Stacked Hourglass Networks for Human Pose Estimation. More details can be found in the [paper](#).

Parameters

- **downsample_times** (int) – Downsample times in a HourglassModule.
- **num_stacks** (int) – Number of HourglassModule modules stacked, 1 for Hourglass-52, 2 for Hourglass-104.
- **stage_channels** (list[int]) – Feature channel of each sub-module in a HourglassModule.
- **stage_blocks** (list[int]) – Number of sub-modules stacked in a HourglassModule.
- **feat_channel** (int) – Feature channel of conv after a HourglassModule.
- **norm_cfg** (dict) – Dictionary to construct and config norm layer.

Example

```
>>> from mmpose.models import HourglassNet
>>> import torch
>>> self = HourglassNet()
>>> self.eval()
>>> inputs = torch.rand(1, 3, 511, 511)
>>> level_outputs = self.forward(inputs)
>>> for level_output in level_outputs:
...     print(tuple(level_output.shape))
(1, 256, 128, 128)
(1, 256, 128, 128)
```

forward(*x*)

Model forward function.

init_weights(*pretrained=None*)

Initialize the weights in backbone.

Parameters pretrained (*str*, *optional*) – Path to pre-trained weights. Defaults to None.

class mmpose.models.backbones.I3D(*in_channels=3*, *expansion=1.0*)

I3D backbone.

Please refer to the [paper](#) for details.

Args: *in_channels* (*int*): Input channels of the backbone, which is decided on the input modality.

expansion (*float*): The multiplier of *in_channels* and *out_channels*. Default: 1.

forward(*x*)

Forward function.

Parameters x (*Tensor* | *tuple[Tensor]*) – *x* could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

class mmpose.models.backbones.LiteHRNet(*extra*, *in_channels=3*, *conv_cfg=None*, *norm_cfg={'type': 'BN'}*, *norm_eval=False*, *with_cp=False*)

Lite-HRNet backbone.

Lite-HRNet: A Lightweight High-Resolution Network.

Code adapted from '<https://github.com/HRNet/Lite-HRNet>'.

Parameters

- **extra** (*dict*) – detailed configuration for each stage of HRNet.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **conv_cfg** (*dict*) – dictionary to construct and config conv layer.
- **norm_cfg** (*dict*) – dictionary to construct and config norm layer.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed.

Example

```
>>> from mmpose.models import LiteHRNet
>>> import torch
>>> extra=dict(
>>>     stem=dict(stem_channels=32, out_channels=32, expand_ratio=1),
>>>     num_stages=3,
>>>     stages_spec=dict(
>>>         num_modules=(2, 4, 2),
>>>         num_branches=(2, 3, 4),
>>>         num_blocks=(2, 2, 2),
>>>         module_type=('LITE', 'LITE', 'LITE'),
```

(continues on next page)

(continued from previous page)

```

>>>         with_fuse=(True, True, True),
>>>         reduce_ratios=(8, 8, 8),
>>>         num_channels=(
>>>             (40, 80),
>>>             (40, 80, 160),
>>>             (40, 80, 160, 320),
>>>         )),
>>>         with_head=False)
>>> self = LiteHRNet(extra, in_channels=1)
>>> self.eval()
>>> inputs = torch.rand(1, 1, 32, 32)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 40, 8, 8)

```

forward(x)

Forward function.

init_weights(pretrained=None)

Initialize the weights in backbone.

Parameters pretrained (str, optional) – Path to pre-trained weights. Defaults to None.

train(mode=True)

Convert the model into training mode.

class mmpose.models.backbones.MSPN(unit_channels=256, num_stages=4, num_units=4, num_blocks=[2, 2, 2, 2], norm_cfg={'type': 'BN'}, res_top_channels=64)

MSPN backbone. Paper ref: Li et al. “Rethinking on Multi-Stage Networks for Human Pose Estimation” (CVPR 2020).

Parameters

- **unit_channels (int)** – Number of Channels in an upsample unit. Default: 256
- **num_stages (int)** – Number of stages in a multi-stage MSPN. Default: 4
- **num_units (int)** – Number of downsample/upsample units in a single-stage network. Default: 4 Note: Make sure num_units == len(self.num_blocks)
- **num_blocks (list)** – Number of bottlenecks in each downsample unit. Default: [2, 2, 2, 2]
- **norm_cfg (dict)** – dictionary to construct and config norm layer. Default: dict(type='BN')
- **res_top_channels (int)** – Number of channels of feature from ResNetTop. Default: 64.

Example

```

>>> from mmpose.models import MSPN
>>> import torch
>>> self = MSPN(num_stages=2,num_units=2,num_blocks=[2,2])
>>> self.eval()
>>> inputs = torch.rand(1, 3, 511, 511)
>>> level_outputs = self.forward(inputs)
>>> for level_output in level_outputs:
...     for feature in level_output:
...         print(tuple(feature.shape))
...
(1, 256, 64, 64)
(1, 256, 128, 128)
(1, 256, 64, 64)
(1, 256, 128, 128)

```

forward(x)

Model forward function.

init_weights(pretrained=None)

Initialize model weights.

```

class mmpose.models.backbones.MobileNetV2(widen_factor=1.0, out_indices=(7, ), frozen_stages=-1,
                                           conv_cfg=None, norm_cfg={'type': 'BN'}, act_cfg={'type':
                                           'ReLU6'}, norm_eval=False, with_cp=False)

```

MobileNetV2 backbone.

Parameters

- **widen_factor** (*float*) – Width multiplier, multiply number of channels in each layer by this amount. Default: 1.0.
- **out_indices** (*None or Sequence[int]*) – Output from which stages. Default: (7,).
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). Default: -1, which means not freezing any parameters.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None, which means using conv2d.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='BN').
- **act_cfg** (*dict*) – Config dict for activation layer. Default: dict(type='ReLU6').
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.

forward(x)

Forward function.

Parameters x (*Tensor / tuple[Tensor]*) – x could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(pretrained=None)

Init backbone weights.

Parameters pretrained (*str* / *None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

make_layer(*out_channels, num_blocks, stride, expand_ratio*)

Stack InvertedResidual blocks to build a layer for MobileNetV2.

Parameters

- **out_channels** (*int*) – out_channels of block.
- **num_blocks** (*int*) – number of blocks.
- **stride** (*int*) – stride of the first block. Default: 1
- **expand_ratio** (*int*) – Expand the number of channels of the hidden layer in InvertedResidual by this ratio. Default: 6.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters mode (*bool*) – whether to set training mode (True) or evaluation mode (False). Default: True.

Returns self

Return type Module

class mmpose.models.backbones.**MobileNetV3**(*arch='small', conv_cfg=None, norm_cfg={'type': 'BN'}, out_indices=(-1), frozen_stages=-1, norm_eval=False, with_cp=False*)

MobileNetV3 backbone.

Parameters

- **arch** (*str*) – Architecture of mobilnetv3, from {small, big}. Default: small.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None, which means using conv2d.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='BN').
- **out_indices** (*None* or *Sequence[int]*) – Output from which stages. Default: (-1,), which means output tensors from final stage.
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). Default: -1, which means not freezing any parameters.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.

forward(*x*)

Forward function.

Parameters x (*Tensor* / *tuple[Tensor]*) – x could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(*pretrained=None*)

Init backbone weights.

Parameters pretrained (*str* / *None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters mode (*bool*) – whether to set training mode (True) or evaluation mode (False).
Default: True.

Returns self

Return type Module

```
class mmpose.models.backbones.PyramidVisionTransformer(pretrain_img_size=224, in_channels=3,
                                                         embed_dims=64, num_stages=4,
                                                         num_layers=[3, 4, 6, 3], num_heads=[1, 2,
                                                         5, 8], patch_sizes=[4, 2, 2, 2], strides=[4, 2,
                                                         2, 2], paddings=[0, 0, 0, 0], sr_ratios=[8, 4,
                                                         2, 1], out_indices=(0, 1, 2, 3),
                                                         mlp_ratios=[8, 8, 4, 4], qkv_bias=True,
                                                         drop_rate=0.0, attn_drop_rate=0.0,
                                                         drop_path_rate=0.1,
                                                         use_abs_pos_embed=True,
                                                         norm_after_stage=False,
                                                         use_conv_ffn=False, act_cfg={'type':
                                                         'GELU'}, norm_cfg={'eps': 1e-06, 'type':
                                                         'LN'}, pretrained=None,
                                                         convert_weights=True, init_cfg=None)
```

Pyramid Vision Transformer (PVT)

Implementation of [Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions](#).

Parameters

- **pretrain_img_size** (*int* / *tuple[int]*) – The size of input image when pretrain. Default: 224.
- **in_channels** (*int*) – Number of input channels. Default: 3.
- **embed_dims** (*int*) – Embedding dimension. Default: 64.
- **num_stags** (*int*) – The num of stages. Default: 4.
- **num_layers** (*Sequence[int]*) – The layer number of each transformer encode layer. Default: [3, 4, 6, 3].
- **num_heads** (*Sequence[int]*) – The attention heads of each transformer encode layer. Default: [1, 2, 5, 8].
- **patch_sizes** (*Sequence[int]*) – The patch_size of each patch embedding. Default: [4, 2, 2, 2].
- **strides** (*Sequence[int]*) – The stride of each patch embedding. Default: [4, 2, 2, 2].
- **paddings** (*Sequence[int]*) – The padding of each patch embedding. Default: [0, 0, 0, 0].
- **sr_ratios** (*Sequence[int]*) – The spatial reduction rate of each transformer encode layer. Default: [8, 4, 2, 1].

- **out_indices** (*Sequence[int] | int*) – Output from which stages. Default: (0, 1, 2, 3).
- **mlp_ratios** (*Sequence[int]*) – The ratio of the mlp hidden dim to the embedding dim of each transformer encode layer. Default: [8, 8, 4, 4].
- **qkv_bias** (*bool*) – Enable bias for qkv if True. Default: True.
- **drop_rate** (*float*) – Probability of an element to be zeroed. Default 0.0.
- **attn_drop_rate** (*float*) – The drop out rate for attention layer. Default 0.0.
- **drop_path_rate** (*float*) – stochastic depth rate. Default 0.1.
- **use_abs_pos_embed** (*bool*) – If True, add absolute position embedding to the patch embedding. Defaults: True.
- **use_conv_ffn** (*bool*) – If True, use Convolutional FFN to replace FFN. Default: False.
- **act_cfg** (*dict*) – The activation config for FFNs. Default: dict(type='GELU').
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='LN').
- **pretrained** (*str, optional*) – model pretrained path. Default: None.
- **convert_weights** (*bool*) – The flag indicates whether the pre-trained model is from the original repo. We may need to convert some keys to make it compatible. Default: True.
- **init_cfg** (*dict or list[dict], optional*) – Initialization config dict. Default: None.

forward(x)

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note: Although the recipe for forward pass needs to be defined within this function, one should call the Module instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

init_weights(pretrained=None)

Initialize the weights.

class mmpose.models.backbones.PyramidVisionTransformerV2(**kwargs)

Implementation of PVTv2: Improved Baselines with Pyramid Vision Transformer.

class mmpose.models.backbones.RSN(unit_channels=256, num_stages=4, num_units=4, num_blocks=[2, 2, 2, 2], num_steps=4, norm_cfg={'type': 'BN'}, res_top_channels=64, expand_times=26)

Residual Steps Network backbone. Paper ref: Cai et al. “Learning Delicate Local Representations for Multi-Person Pose Estimation” (ECCV 2020).

Parameters

- **unit_channels** (*int*) – Number of Channels in an upsample unit. Default: 256
- **num_stages** (*int*) – Number of stages in a multi-stage RSN. Default: 4
- **num_units** (*int*) – Number of downsample/upsample units in a single-stage RSN. Default: 4 Note: Make sure num_units == len(self.num_blocks)
- **num_blocks** (*list*) – Number of RSBs (Residual Steps Block) in each downsample unit. Default: [2, 2, 2, 2]
- **num_steps** (*int*) – Number of steps in a RSB. Default: 4

- **norm_cfg** (*dict*) – dictionary to construct and config norm layer. Default: dict(type='BN')
- **res_top_channels** (*int*) – Number of channels of feature from ResNet_top. Default: 64.
- **expand_times** (*int*) – Times by which the in_channels are expanded in RSB. Default:26.

Example

```
>>> from mmpose.models import RSN
>>> import torch
>>> self = RSN(num_stages=2,num_units=2,num_blocks=[2,2])
>>> self.eval()
>>> inputs = torch.rand(1, 3, 511, 511)
>>> level_outputs = self.forward(inputs)
>>> for level_output in level_outputs:
...     for feature in level_output:
...         print(tuple(feature.shape))
...
(1, 256, 64, 64)
(1, 256, 128, 128)
(1, 256, 64, 64)
(1, 256, 128, 128)
```

forward(*x*)

Model forward function.

init_weights(*pretrained=None*)

Initialize model weights.

```
class mmpose.models.backbones.RegNet(arch, in_channels=3, stem_channels=32, base_channels=32,
                                     strides=(2, 2, 2, 2), dilations=(1, 1, 1, 1), out_indices=(3),
                                     style='pytorch', deep_stem=False, avg_down=False,
                                     frozen_stages=- 1, conv_cfg=None, norm_cfg={'requires_grad':
                                     True, 'type': 'BN'}, norm_eval=False, with_cp=False,
                                     zero_init_residual=True)
```

RegNet backbone.

More details can be found in [paper](#) .

Parameters

- **arch** (*dict*) – The parameter of RegNets. - w0 (int): initial width - wa (float): slope of width - wm (float): quantization parameter to quantize the width - depth (int): depth of the backbone - group_w (int): width of group - bot_mul (float): bottleneck ratio, i.e. expansion of bottleneck.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage.
- **base_channels** (*int*) – Base channels after stem layer.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **dilations** (*Sequence[int]*) – Dilation of each stage.
- **out_indices** (*Sequence[int]*) – Output from which stages.
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer. Default: “pytorch”.

- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). -1 means not freezing any parameters. Default: -1.
- **norm_cfg** (*dict*) – dictionary to construct and config norm layer. Default: dict(type='BN', requires_grad=True).
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

Example

```
>>> from mmpose.models import RegNet
>>> import torch
>>> self = RegNet(
    arch=dict(
        w0=88,
        wa=26.31,
        wm=2.25,
        group_w=48,
        depth=25,
        bot_mul=1.0),
    out_indices=(0, 1, 2, 3))
>>> self.eval()
>>> inputs = torch.rand(1, 3, 32, 32)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 96, 8, 8)
(1, 192, 4, 4)
(1, 432, 2, 2)
(1, 1008, 1, 1)
```

adjust_width_group(*widths, bottleneck_ratio, groups*)

Adjusts the compatibility of widths and groups.

Parameters

- **widths** (*list[int]*) – Width of each stage.
- **bottleneck_ratio** (*float*) – Bottleneck ratio.
- **groups** (*int*) – number of groups in each stage

Returns The adjusted widths and groups of each stage.

Return type tuple(list)

forward(*x*)

Forward function.

static generate_regnet(*initial_width, width_slope, width_parameter, depth, divisor=8*)

Generates per block width from RegNet parameters.

Parameters

- **initial_width** (*[int]*) – Initial width of the backbone
- **width_slope** (*[float]*) – Slope of the quantized linear function
- **width_parameter** (*[int]*) – Parameter used to quantize the width.
- **depth** (*[int]*) – Depth of the backbone.
- **divisor** (*int, optional*) – The divisor of channels. Defaults to 8.

Returns

return a list of widths of each stage and the number of stages

Return type list, int

get_stages_from_blocks(*widths*)

Gets widths/stage_blocks of network at each stage.

Parameters **widths** (*list[int]*) – Width in each stage.

Returns width and depth of each stage

Return type tuple(list)

static quantize_float(*number, divisor*)

Converts a float to closest non-zero int divisible by divisor.

Parameters

- **number** (*int*) – Original number to be quantized.
- **divisor** (*int*) – Divisor used to quantize the number.

Returns quantized number that is divisible by divisor.

Return type int

class mmpose.models.backbones.**ResNeSt**(*depth, groups=1, width_per_group=4, radix=2, reduction_factor=4, avg_down_stride=True, **kwargs*)

ResNeSt backbone.

Please refer to the [paper](#) for details.

Parameters

- **depth** (*int*) – Network depth, from {50, 101, 152, 200}.
- **groups** (*int*) – Groups of conv2 in Bottleneck. Default: 32.
- **width_per_group** (*int*) – Width per group of conv2 in Bottleneck. Default: 4.
- **radix** (*int*) – Radix of SpltAtConv2d. Default: 2
- **reduction_factor** (*int*) – Reduction factor of SplitAttentionConv2d. Default: 4.
- **avg_down_stride** (*bool*) – Whether to use average pool for stride in Bottleneck. Default: True.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **stem_channels** (*int*) – Output channels of the stem layer. Default: 64.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).

- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottleneck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* | *None*) – The config dict for conv layers. Default: None.
- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

make_res_layer(***kwargs*)

Make a ResLayer.

class mmpose.models.backbones.**ResNeXt**(*depth*, *groups*=32, *width_per_group*=4, ***kwargs*)

ResNeXt backbone.

Please refer to the [paper](#) for details.

Parameters

- **depth** (*int*) – Network depth, from {50, 101, 152}.
- **groups** (*int*) – Groups of conv2 in Bottleneck. Default: 32.
- **width_per_group** (*int*) – Width per group of conv2 in Bottleneck. Default: 4.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **stem_channels** (*int*) – Output channels of the stem layer. Default: 64.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).
- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.

- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottle-neck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* / *None*) – The config dict for conv layers. Default: None.
- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

make_res_layer(***kwargs*)

Make a ResLayer.

```
class mmpose.models.backbones.ResNet(depth, in_channels=3, stem_channels=64, base_channels=64,
                                     expansion=None, num_stages=4, strides=(1, 2, 2, 2), dilations=(1,
                                     1, 1, 1), out_indices=(3), style='pytorch', deep_stem=False,
                                     avg_down=False, frozen_stages=-1, conv_cfg=None,
                                     norm_cfg={'requires_grad': True, 'type': 'BN'}, norm_eval=False,
                                     with_cp=False, zero_init_residual=True)
```

ResNet backbone.

Please refer to the [paper](#) for details.

Parameters

- **depth** (*int*) – Network depth, from {18, 34, 50, 101, 152}.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **stem_channels** (*int*) – Output channels of the stem layer. Default: 64.
- **base_channels** (*int*) – Middle channels of the first stage. Default: 64.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).
- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottle-neck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* / *None*) – The config dict for conv layers. Default: None.

- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

Example

```
>>> from mmpose.models import ResNet
>>> import torch
>>> self = ResNet(depth=18, out_indices=(0, 1, 2, 3))
>>> self.eval()
>>> inputs = torch.rand(1, 3, 32, 32)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 64, 8, 8)
(1, 128, 4, 4)
(1, 256, 2, 2)
(1, 512, 1, 1)
```

forward(*x*)

Forward function.

init_weights(*pretrained=None*)

Initialize the weights in backbone.

Parameters **pretrained** (*str, optional*) – Path to pre-trained weights. Defaults to None.

make_res_layer(***kwargs*)

Make a ResLayer.

property norm1

the normalization layer named “norm1”

Type nn.Module

train(*mode=True*)

Convert the model into training mode.

class mmpose.models.backbones.**ResNetV1d**(***kwargs*)

ResNetV1d variant described in [Bag of Tricks](#).

Compared with default ResNet(ResNetV1b), ResNetV1d replaces the 7x7 conv in the input stem with three 3x3 convs. And in the downsampling block, a 2x2 avg_pool with stride 2 is added before conv, whose stride is changed to 1.

class mmpose.models.backbones.**SCNet**(*depth, **kwargs*)

SCNet backbone.

Improving Convolutional Networks with Self-Calibrated Convolutions, Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Changhu Wang, Jiashi Feng, IEEE CVPR, 2020. <http://mftp.mmcheng.net/Papers/20cvprSCNet.pdf>

Parameters

- **depth** (*int*) – Depth of scnet, from {50, 101}.
- **in_channels** (*int*) – Number of input image channels. Normally 3.
- **base_channels** (*int*) – Number of base channels of hidden layer.
- **num_stages** (*int*) – SCNet stages, normally 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage.
- **dilations** (*Sequence[int]*) – Dilation of each stage.
- **out_indices** (*Sequence[int]*) – Output from which stages.
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottle-neck.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters.
- **norm_cfg** (*dict*) – Dictionary to construct and config norm layer.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity.

Example

```
>>> from mmpose.models import SCNet
>>> import torch
>>> self = SCNet(depth=50, out_indices=(0, 1, 2, 3))
>>> self.eval()
>>> inputs = torch.rand(1, 3, 224, 224)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 256, 56, 56)
(1, 512, 28, 28)
(1, 1024, 14, 14)
(1, 2048, 7, 7)
```

class mmpose.models.backbones.**SEResNeXt**(*depth*, *groups*=32, *width_per_group*=4, ***kwargs*)
SEResNeXt backbone.

Please refer to the [paper](#) for details.

Parameters

- **depth** (*int*) – Network depth, from {50, 101, 152}.
- **groups** (*int*) – Groups of conv2 in Bottleneck. Default: 32.

- **width_per_group** (*int*) – Width per group of conv2 in Bottleneck. Default: 4.
- **se_ratio** (*int*) – Squeeze ratio in SELayer. Default: 16.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **stem_channels** (*int*) – Output channels of the stem layer. Default: 64.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).
- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottleneck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* | *None*) – The config dict for conv layers. Default: None.
- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

Example

```
>>> from mmpose.models import SEResNeXt
>>> import torch
>>> self = SEResNet(depth=50, out_indices=(0, 1, 2, 3))
>>> self.eval()
>>> inputs = torch.rand(1, 3, 224, 224)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 256, 56, 56)
(1, 512, 28, 28)
(1, 1024, 14, 14)
(1, 2048, 7, 7)
```

make_res_layer (***kwargs*)
Make a ResLayer.

class `mmpose.models.backbones.SEResNet`(*depth*, *se_ratio*=16, ***kwargs*)
 SEResNet backbone.

Please refer to the [paper](#) for details.

Parameters

- **depth** (*int*) – Network depth, from {50, 101, 152}.
- **se_ratio** (*int*) – Squeeze ratio in SELayer. Default: 16.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **stem_channels** (*int*) – Output channels of the stem layer. Default: 64.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).
- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottle-neck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* | *None*) – The config dict for conv layers. Default: None.
- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.

Example

```
>>> from mmpose.models import SEResNet
>>> import torch
>>> self = SEResNet(depth=50, out_indices=(0, 1, 2, 3))
>>> self.eval()
>>> inputs = torch.rand(1, 3, 224, 224)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 256, 56, 56)
(1, 512, 28, 28)
```

(continues on next page)

(continued from previous page)

```
(1, 1024, 14, 14)
(1, 2048, 7, 7)
```

make_res_layer(**kwargs)

Make a ResLayer.

```
class mmpose.models.backbones.ShuffleNetV1(groups=3, widen_factor=1.0, out_indices=(2),
                                           frozen_stages=-1, conv_cfg=None, norm_cfg={'type':
                                           'BN'}, act_cfg={'type': 'ReLU'}, norm_eval=False,
                                           with_cp=False)
```

ShuffleNetV1 backbone.

Parameters

- **groups** (*int, optional*) – The number of groups to be used in grouped 1x1 convolutions in each ShuffleUnit. Default: 3.
- **widen_factor** (*float, optional*) – Width multiplier - adjusts the number of channels in each layer by this amount. Default: 1.0.
- **out_indices** (*Sequence[int]*) – Output from which stages. Default: (2,)
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). Default: -1, which means not freezing any parameters.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None, which means using conv2d.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='BN').
- **act_cfg** (*dict*) – Config dict for activation layer. Default: dict(type='ReLU').
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.

forward(x)

Forward function.

Parameters **x** (*Tensor | tuple[Tensor]*) – x could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(pretrained=None)

Init backbone weights.

Parameters **pretrained** (*str | None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

make_layer(out_channels, num_blocks, first_block=False)

Stack ShuffleUnit blocks to make a layer.

Parameters

- **out_channels** (*int*) – out_channels of the block.
- **num_blocks** (*int*) – Number of blocks.
- **first_block** (*bool, optional*) – Whether is the first ShuffleUnit of a sequential ShuffleUnits. Default: False, which means using the grouped 1x1 convolution.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters *mode* (*bool*) – whether to set training mode (True) or evaluation mode (False).

Default: True.

Returns *self*

Return type Module

class `mmpose.models.backbones.ShuffleNetV2`(*widen_factor=1.0, out_indices=(3), frozen_stages=-1, conv_cfg=None, norm_cfg={'type': 'BN'}, act_cfg={'type': 'ReLU'}, norm_eval=False, with_cp=False*)

ShuffleNetV2 backbone.

Parameters

- **widen_factor** (*float*) – Width multiplier - adjusts the number of channels in each layer by this amount. Default: 1.0.
- **out_indices** (*Sequence[int]*) – Output from which stages. Default: (0, 1, 2, 3).
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). Default: -1, which means not freezing any parameters.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None, which means using conv2d.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='BN').
- **act_cfg** (*dict*) – Config dict for activation layer. Default: dict(type='ReLU').
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.

forward(*x*)

Forward function.

Parameters *x* (*Tensor | tuple[Tensor]*) – *x* could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(*pretrained=None*)

Init backbone weights.

Parameters *pretrained* (*str | None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters *mode* (*bool*) – whether to set training mode (True) or evaluation mode (False).

Default: True.

Returns *self*

Return type Module

```
class mmpose.models.backbones.SwinTransformer(pretrain_img_size=224, in_channels=3,
                                              embed_dims=96, patch_size=4, window_size=7,
                                              mlp_ratio=4, depths=(2, 2, 6, 2), num_heads=(3, 6, 12,
                                              24), strides=(4, 2, 2, 2), out_indices=(0, 1, 2, 3),
                                              qkv_bias=True, qk_scale=None, patch_norm=True,
                                              drop_rate=0.0, attn_drop_rate=0.0,
                                              drop_path_rate=0.1, use_abs_pos_embed=False,
                                              act_cfg={'type': 'GELU'}, norm_cfg={'type': 'LN'},
                                              with_cp=False, convert_weights=False, frozen_stages=-1)
```

Swin Transformer A PyTorch implement of : *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows* -

<https://arxiv.org/abs/2103.14030>

Inspiration from <https://github.com/microsoft/Swin-Transformer>

Parameters

- **pretrain_img_size** (*int* | *tuple[int]*) – The size of input image when pretrain. Defaults: 224.
- **in_channels** (*int*) – The num of input channels. Defaults: 3.
- **embed_dims** (*int*) – The feature dimension. Default: 96.
- **patch_size** (*int* | *tuple[int]*) – Patch size. Default: 4.
- **window_size** (*int*) – Window size. Default: 7.
- **mlp_ratio** (*int*) – Ratio of mlp hidden dim to embedding dim. Default: 4.
- **depths** (*tuple[int]*) – Depths of each Swin Transformer stage. Default: (2, 2, 6, 2).
- **num_heads** (*tuple[int]*) – Parallel attention heads of each Swin Transformer stage. Default: (3, 6, 12, 24).
- **strides** (*tuple[int]*) – The patch merging or patch embedding stride of each Swin Transformer stage. (In swin, we set kernel size equal to stride.) Default: (4, 2, 2, 2).
- **out_indices** (*tuple[int]*) – Output from which stages. Default: (0, 1, 2, 3).
- **qkv_bias** (*bool*, *optional*) – If True, add a learnable bias to query, key, value. Default: True
- **qk_scale** (*float* | *None*, *optional*) – Override default qk scale of head_dim ** -0.5 if set. Default: None.
- **patch_norm** (*bool*) – If add a norm layer for patch embed and patch merging. Default: True.
- **drop_rate** (*float*) – Dropout rate. Defaults: 0.
- **attn_drop_rate** (*float*) – Attention dropout rate. Default: 0.
- **drop_path_rate** (*float*) – Stochastic depth rate. Defaults: 0.1.
- **use_abs_pos_embed** (*bool*) – If True, add absolute position embedding to the patch embedding. Defaults: False.
- **act_cfg** (*dict*) – Config dict for activation layer. Default: dict(type='LN').

- **norm_cfg** (*dict*) – Config dict for normalization layer at output of backbone. Defaults: dict(type='LN').
- **with_cp** (*bool, optional*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **pretrained** (*str, optional*) – model pretrained path. Default: None.
- **convert_weights** (*bool*) – The flag indicates whether the pre-trained model is from the original repo. We may need to convert some keys to make it compatible. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). Default: -1 (-1 means not freezing any parameters).

forward(*x*)

Forward function.

Parameters *x* (*Tensor | tuple[Tensor]*) – *x* could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(*pretrained=None*)

Initialize the weights in backbone.

Parameters **pretrained** (*str, optional*) – Path to pre-trained weights. Defaults to None.

train(*mode=True*)

Convert the model into training mode while keep layers freezed.

```
class mmpose.models.backbones.TCFormer(in_channels=3, embed_dims=[64, 128, 256, 512],
                                       num_heads=[1, 2, 4, 8], mlp_ratios=[4, 4, 4, 4], qkv_bias=True,
                                       qk_scale=None, drop_rate=0.0, attn_drop_rate=0.0,
                                       drop_path_rate=0.0, norm_cfg={'eps': 1e-06, 'type': 'LN'},
                                       num_layers=[3, 4, 6, 3], sr_ratios=[8, 4, 2, 1], num_stages=4,
                                       pretrained=None, k=5, sample_ratios=[0.25, 0.25, 0.25],
                                       return_map=False, convert_weights=True)
```

Token Clustering Transformer (TCFormer)

Implementation of *Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering Transformer*
<<https://arxiv.org/abs/2204.08680>>

Args: *in_channels* (int): Number of input channels. Default: 3. *embed_dims* (list[int]): Embedding dimension. Default:

[64, 128, 256, 512].

num_heads (Sequence[int]): The attention heads of each transformer encode layer. Default: [1, 2, 5, 8].

mlp_ratios (Sequence[int]): The ratio of the mlp hidden dim to the embedding dim of each transformer block.

qkv_bias (bool): Enable bias for qkv if True. Default: True. *qk_scale* (float | None, optional): Override default qk scale of

head_dim ** -0.5 if set. Default: None.

drop_rate (float): Probability of an element to be zeroed. Default 0.0.

attn_drop_rate (float): The drop out rate for attention layer. Default 0.0.

drop_path_rate (float): stochastic depth rate. Default 0. *norm_cfg* (dict): Config dict for normalization layer.

Default: dict(type='LN', eps=1e-6).

num_layers (Sequence[int]): The layer number of each transformer encode layer. Default: [3, 4, 6, 3].

sr_ratios (Sequence[int]): The spatial reduction rate of each transformer block. Default: [8, 4, 2, 1].

num_stages (int): The num of stages. Default: 4. **pretrained** (str, optional): model pretrained path. Default: None. **k** (int): number of the nearest neighbor used for local density. **sample_ratios** (list[float]): The sample ratios of CTM modules.

Default: [0.25, 0.25, 0.25]

return_map (bool): If True, transfer dynamic tokens to feature map at last. Default: False

convert_weights (bool): The flag indicates whether the pre-trained model is from the original repo. We may need to convert some keys to make it compatible. Default: True.

forward(x)

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note: Although the recipe for forward pass needs to be defined within this function, one should call the Module instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

```
class mmpose.models.backbones.TCN(in_channels, stem_channels=1024, num_blocks=2, kernel_sizes=(3, 3, 3), dropout=0.25, causal=False, residual=True, use_stride_conv=False, conv_cfg={'type': 'Conv1d'}, norm_cfg={'type': 'BN1d'}, max_norm=None)
```

TCN backbone.

Temporal Convolutional Networks. More details can be found in the [paper](#).

Parameters

- **in_channels** (int) – Number of input channels, which equals to num_keypoints * num_features.
- **stem_channels** (int) – Number of feature channels. Default: 1024.
- **num_blocks** (int) – Number of basic temporal convolutional blocks. Default: 2.
- **kernel_sizes** (Sequence[int]) – Sizes of the convolving kernel of each basic block. Default: (3, 3, 3).
- **dropout** (float) – Dropout rate. Default: 0.25.
- **causal** (bool) – Use causal convolutions instead of symmetric convolutions (for real-time applications). Default: False.
- **residual** (bool) – Use residual connection. Default: True.
- **use_stride_conv** (bool) – Use TCN backbone optimized for single-frame batching, i.e. where batches have input length = receptive field, and output length = 1. This implementation replaces dilated convolutions with strided convolutions to avoid generating unused intermediate results. The weights are interchangeable with the reference implementation. Default: False

- **conv_cfg** (*dict*) – dictionary to construct and config conv layer. Default: `dict(type='Conv1d')`.
- **norm_cfg** (*dict*) – dictionary to construct and config norm layer. Default: `dict(type='BN1d')`.
- **max_norm** (*float/None*) – if not `None`, the weight of convolution layers will be clipped to have a maximum norm of `max_norm`.

Example

```
>>> from mmpose.models import TCN
>>> import torch
>>> self = TCN(in_channels=34)
>>> self.eval()
>>> inputs = torch.rand(1, 34, 243)
>>> level_outputs = self.forward(inputs)
>>> for level_out in level_outputs:
...     print(tuple(level_out.shape))
(1, 1024, 235)
(1, 1024, 217)
```

forward(*x*)

Forward function.

init_weights(*pretrained=None*)

Initialize the weights.

class mmpose.models.backbones.V2VNet(*input_channels, output_channels, mid_channels=32*)
V2VNet.

Please refer to the *paper* <<https://arxiv.org/abs/1711.07399>> for details.

Parameters

- **input_channels** (*int*) – Number of channels of the input feature volume.
- **output_channels** (*int*) – Number of channels of the output volume.
- **mid_channels** (*int*) – Input and output channels of the encoder-decoder block.

forward(*x*)

Forward function.

class mmpose.models.backbones.VGG(*depth, num_classes=-1, num_stages=5, dilations=(1, 1, 1, 1, 1), out_indices=None, frozen_stages=-1, conv_cfg=None, norm_cfg=None, act_cfg={'type': 'ReLU'}, norm_eval=False, ceil_mode=False, with_last_pool=True*)

VGG backbone.

Parameters

- **depth** (*int*) – Depth of vgg, from {11, 13, 16, 19}.
- **with_norm** (*bool*) – Use BatchNorm or not.
- **num_classes** (*int*) – number of classes for classification.
- **num_stages** (*int*) – VGG stages, normally 5.

- **dilations** (*Sequence[int]*) – Dilation of each stage.
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. When it is None, the default behavior depends on whether num_classes is specified. If num_classes ≤ 0, the default value is (4,), outputting the last feature map before classifier. If num_classes > 0, the default value is (5,), outputting the classification score. Default: None.
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). -1 means not freezing any parameters.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **ceil_mode** (*bool*) – Whether to use ceil_mode of MaxPool. Default: False.
- **with_last_pool** (*bool*) – Whether to keep the last pooling before classifier. Default: True.

forward(*x*)

Forward function.

Parameters *x* (*Tensor* / *tuple[Tensor]*) – *x* could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(*pretrained=None*)

Init backbone weights.

Parameters *pretrained* (*str* / *None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters *mode* (*bool*) – whether to set training mode (True) or evaluation mode (False). Default: True.

Returns *self*

Return type Module

```
class mmpose.models.backbones.ViPNAS_MobileNetV3(wid=[16, 16, 24, 40, 80, 112, 160], expan=[None, 1,
5, 4, 5, 5, 6], dep=[None, 1, 4, 4, 4, 4, 4], ks=[3, 3,
7, 7, 5, 7, 5], group=[None, 8, 120, 20, 100, 280,
240], att=[None, True, True, False, True, True,
True], stride=[2, 1, 2, 2, 2, 1, 2], act=['HSwish',
'ReLU', 'ReLU', 'ReLU', 'HSwish', 'HSwish',
'HSwish'], conv_cfg=None, norm_cfg={'type':
'BN'}, frozen_stages=-1, norm_eval=False,
with_cp=False)
```

ViPNAS_MobileNetV3 backbone.

“ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search” More details can be found in the [paper](#).

Parameters

- **wid** (*list(int)*) – Searched width config for each stage.

- **expan** (*list(int)*) – Searched expansion ratio config for each stage.
- **dep** (*list(int)*) – Searched depth config for each stage.
- **ks** (*list(int)*) – Searched kernel size config for each stage.
- **group** (*list(int)*) – Searched group number config for each stage.
- **att** (*list(bool)*) – Searched attention config for each stage.
- **stride** (*list(int)*) – Stride config for each stage.
- **act** (*list(dict)*) – Activation config for each stage.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None, which means using conv2d.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: dict(type='BN').
- **frozen_stages** (*int*) – Stages to be frozen (all param fixed). Default: -1, which means not freezing any parameters.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.

forward(*x*)

Forward function.

Parameters *x* (*Tensor* / *tuple[Tensor]*) – *x* could be a torch.Tensor or a tuple of torch.Tensor, containing input data for forward computation.

init_weights(*pretrained=None*)

Init backbone weights.

Parameters **pretrained** (*str* / *None*) – If pretrained is a string, then it initializes backbone weights by loading the pretrained checkpoint. If pretrained is None, then it follows default initializer or customized initializer in subclasses.

train(*mode=True*)

Sets the module in training mode.

This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. Dropout, BatchNorm, etc.

Parameters **mode** (*bool*) – whether to set training mode (True) or evaluation mode (False). Default: True.

Returns self

Return type Module

```
class mmpose.models.backbones.ViPNAS_ResNet(depth, in_channels=3, num_stages=4, strides=(1, 2, 2, 2),
                                             dilations=(1, 1, 1, 1), out_indices=(3), style='pytorch',
                                             deep_stem=False, avg_down=False, frozen_stages=-1,
                                             conv_cfg=None, norm_cfg={'requires_grad': True, 'type':
                                             'BN'}, norm_eval=False, with_cp=False,
                                             zero_init_residual=True, wid=[48, 80, 160, 304, 608],
                                             expan=[None, 1, 1, 1, 1], dep=[None, 4, 6, 7, 3], ks=[7, 3,
                                             5, 5, 5], group=[None, 16, 16, 16, 16], att=[None, True,
                                             False, True, True])
```

ViPNAS_ResNet backbone.

“ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search” More details can be found in the [paper](#).

Parameters

- **depth** (*int*) – Network depth, from {18, 34, 50, 101, 152}.
- **in_channels** (*int*) – Number of input image channels. Default: 3.
- **num_stages** (*int*) – Stages of the network. Default: 4.
- **strides** (*Sequence[int]*) – Strides of the first block of each stage. Default: (1, 2, 2, 2).
- **dilations** (*Sequence[int]*) – Dilation of each stage. Default: (1, 1, 1, 1).
- **out_indices** (*Sequence[int]*) – Output from which stages. If only one stage is specified, a single tensor (feature map) is returned, otherwise multiple stages are specified, a tuple of tensors will be returned. Default: (3,).
- **style** (*str*) – *pytorch* or *caffe*. If set to “pytorch”, the stride-two layer is the 3x3 conv layer, otherwise the stride-two layer is the first 1x1 conv layer.
- **deep_stem** (*bool*) – Replace 7x7 conv in input stem with 3 3x3 conv. Default: False.
- **avg_down** (*bool*) – Use AvgPool instead of stride conv when downsampling in the bottle-neck. Default: False.
- **frozen_stages** (*int*) – Stages to be frozen (stop grad and set eval mode). -1 means not freezing any parameters. Default: -1.
- **conv_cfg** (*dict* | *None*) – The config dict for conv layers. Default: None.
- **norm_cfg** (*dict*) – The config dict for norm layers.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **with_cp** (*bool*) – Use checkpoint or not. Using checkpoint will save some memory while slowing down the training speed. Default: False.
- **zero_init_residual** (*bool*) – Whether to use zero init for last norm layer in resblocks to let them behave as identity. Default: True.
- **wid** (*list(int)*) – Searched width config for each stage.
- **expan** (*list(int)*) – Searched expansion ratio config for each stage.
- **dep** (*list(int)*) – Searched depth config for each stage.
- **ks** (*list(int)*) – Searched kernel size config for each stage.
- **group** (*list(int)*) – Searched group number config for each stage.
- **att** (*list(bool)*) – Searched attention config for each stage.

forward(*x*)

Forward function.

init_weights(*pretrained=None*)

Initialize model weights.

make_res_layer(***kwargs*)

Make a ViPNAS ResLayer.

property norm1

the normalization layer named “norm1”

Type `nn.Module`

train(*mode=True*)

Convert the model into training mode.

49.2 necks

```
class mmpose.models.necks.FPN(in_channels, out_channels, num_outs, start_level=0, end_level=-1,  
                             add_extra_convs=False, relu_before_extra_convs=False,  
                             no_norm_on_lateral=False, conv_cfg=None, norm_cfg=None,  
                             act_cfg=None, upsample_cfg={'mode': 'nearest'})
```

Feature Pyramid Network.

This is an implementation of paper [Feature Pyramid Networks for Object Detection](#).

Parameters

- **in_channels** (*list[int]*) – Number of input channels per scale.
- **out_channels** (*int*) – Number of output channels (used at each scale).
- **num_outs** (*int*) – Number of output scales.
- **start_level** (*int*) – Index of the start input backbone level used to build the feature pyramid. Default: 0.
- **end_level** (*int*) – Index of the end input backbone level (exclusive) to build the feature pyramid. Default: -1, which means the last level.
- **add_extra_convs** (*bool | str*) – If *bool*, it decides whether to add conv layers on top of the original feature maps. Default to *False*. If *True*, it is equivalent to *add_extra_convs='on_input'*. If *str*, it specifies the source feature map of the extra convs. Only the following options are allowed
 - *'on_input'*: Last feat map of neck inputs (i.e. backbone feature).
 - *'on_lateral'*: Last feature map after lateral convs.
 - *'on_output'*: The last output feature map after fpn convs.
- **relu_before_extra_convs** (*bool*) – Whether to apply relu before the extra conv. Default: *False*.
- **no_norm_on_lateral** (*bool*) – Whether to apply norm on lateral. Default: *False*.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: *None*.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: *None*.
- **act_cfg** (*dict*) – Config dict for activation layer in *ConvModule*. Default: *None*.
- **upsample_cfg** (*dict*) – Config dict for interpolate layer. Default: *dict(mode='nearest')*.

Example

```

>>> import torch
>>> in_channels = [2, 3, 5, 7]
>>> scales = [340, 170, 84, 43]
>>> inputs = [torch.rand(1, c, s, s)
...           for c, s in zip(in_channels, scales)]
>>> self = FPN(in_channels, 11, len(in_channels)).eval()
>>> outputs = self.forward(inputs)
>>> for i in range(len(outputs)):
...     print(f'outputs[{i}].shape = {outputs[i].shape}')
outputs[0].shape = torch.Size([1, 11, 340, 340])
outputs[1].shape = torch.Size([1, 11, 170, 170])
outputs[2].shape = torch.Size([1, 11, 84, 84])
outputs[3].shape = torch.Size([1, 11, 43, 43])

```

forward(inputs)

Forward function.

init_weights()

Initialize model weights.

class mmpose.models.necks.GlobalAveragePooling

Global Average Pooling neck.

Note that we use *view* to remove extra channel after pooling. We do not use *squeeze* as it will also remove the batch dimension when the tensor has a batch dimension of size 1, which can lead to unexpected errors.

forward(inputs)

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note: Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

```

class mmpose.models.necks.MTA(in_channels=[64, 128, 256, 512], out_channels=128, num_outs=4,
                               start_level=0, end_level=-1, add_extra_convs=False,
                               relu_before_extra_convs=False, no_norm_on_lateral=False,
                               conv_cfg=None, norm_cfg=None, act_cfg=None, num_heads=[2, 2, 2, 2],
                               mlp_ratios=[4, 4, 4, 4], sr_ratios=[8, 4, 2, 1], qkv_bias=True,
                               qk_scale=None, drop_rate=0.0, attn_drop_rate=0.0, drop_path_rate=0.0,
                               transformer_norm_cfg={'type': 'LN'}, use_sr_conv=False)

```

Multi-stage Token feature Aggregation (MTA) module in TCFormer.

Parameters

- **in_channels** (*list[int]*) – Number of input channels per stage. Default: [64, 128, 256, 512].
- **out_channels** (*int*) – Number of output channels (used at each scale).
- **num_outs** (*int*) – Number of output scales. Default: 4.
- **start_level** (*int*) – Index of the start input backbone level used to build the feature pyramid. Default: 0.

- **end_level** (*int*) – Index of the end input backbone level (exclusive) to build the feature pyramid. Default: -1, which means the last level.
- **add_extra_convs** (*bool | str*) – If bool, it decides whether to add conv layers on top of the original feature maps. Default to False. If True, it is equivalent to *add_extra_convs='on_input'*. If str, it specifies the source feature map of the extra convs. Only the following options are allowed - 'on_input': Last feat map of neck inputs (i.e. backbone feature). - 'on_output': The last output feature map after fpn convs.
- **relu_before_extra_convs** (*bool*) – Whether to apply relu before the extra conv. Default: False.
- **no_norm_on_lateral** (*bool*) – Whether to apply norm on lateral. Default: False.
- **conv_cfg** (*dict*) – Config dict for convolution layer. Default: None.
- **norm_cfg** (*dict*) – Config dict for normalization layer. Default: None.
- **act_cfg** (*dict*) – Config dict for activation layer in ConvModule.
- **num_heads** (*Sequence[int]*) – The attention heads of each transformer block. Default: [2, 2, 2, 2].
- **mlp_ratios** (*Sequence[int]*) – The ratio of the mlp hidden dim to the embedding dim of each transformer block.
- **sr_ratios** (*Sequence[int]*) – The spatial reduction rate of each transformer block. Default: [8, 4, 2, 1].
- **qkv_bias** (*bool*) – Enable bias for qkv if True. Default: True.
- **qk_scale** (*float | None, optional*) – Override default qk scale of head_dim ** -0.5 if set. Default: None.
- **drop_rate** (*float*) – Probability of an element to be zeroed. Default 0.0.
- **attn_drop_rate** (*float*) – The drop out rate for attention layer. Default 0.0.
- **drop_path_rate** (*float*) – stochastic depth rate. Default 0.
- **transformer_norm_cfg** (*dict*) – Config dict for normalization layer in transformer blocks. Default: dict(type='LN').
- **use_sr_conv** (*bool*) – If True, use a conv layer for spatial reduction. If False, use a pooling process for spatial reduction. Defaults: False.

forward(*inputs*)

Forward function.

init_weights()

Initialize the weights.

```
class mmpose.models.necks.PoseWarperNeck(in_channels, out_channels, inner_channels,
                                         deform_groups=17, dilations=(3, 6, 12, 18, 24),
                                         trans_conv_kernel=1, res_blocks_cfg=None, offsets_kernel=3,
                                         deform_conv_kernel=3, in_index=0, input_transform=None,
                                         freeze_trans_layer=True, norm_eval=False, im2col_step=80)
```

PoseWarper neck.

“Learning temporal pose estimation from sparsely-labeled videos”.

Parameters

- **in_channels** (*int*) – Number of input channels from backbone

- **out_channels** (*int*) – Number of output channels
- **inner_channels** (*int*) – Number of intermediate channels of the res block
- **deform_groups** (*int*) – Number of groups in the deformable conv
- **dilations** (*list/tuple*) – different dilations of the offset conv layers
- **trans_conv_kernel** (*int*) – the kernel of the trans conv layer, which is used to get heatmap from the output of backbone. Default: 1
- **res_blocks_cfg** (*dict/None*) – config of residual blocks. If None, use the default values. If not None, it should contain the following keys:
 - **block** (*str*): the type of residual block, Default: 'BASIC'.
 - **num_blocks** (*int*): the number of blocks, Default: 20.
- **offsets_kernel** (*int*) – the kernel of offset conv layer.
- **deform_conv_kernel** (*int*) – the kernel of deformable conv layer.
- **in_index** (*int/Sequence[int]*) – Input feature index. Default: 0
- **input_transform** (*str/None*) – Transformation type of input features. Options: 'resize_concat', 'multiple_select', None. Default: None.
 - 'resize_concat': Multiple feature maps will be resize to the same size as first one and then concat together. Usually used in FCN head of HRNet.
 - 'multiple_select': Multiple feature maps will be bundle into a list and passed into decode head.
 - None: Only one select feature map is allowed.
- **freeze_trans_layer** (*bool*) – Whether to freeze the transition layer (stop grad and set eval mode). Default: True.
- **norm_eval** (*bool*) – Whether to set norm layers to eval mode, namely, freeze running stats (mean and var). Note: Effect on Batch Norm and its variants only. Default: False.
- **im2col_step** (*int*) – the argument *im2col_step* in deformable conv, Default: 80.

forward(*inputs, frame_weight*)

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note: Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

train(*mode=True*)

Convert the model into training mode.

49.3 detectors

class `mmpose.models.detectors.AssociativeEmbedding`(*backbone*, *keypoint_head*=None, *train_cfg*=None, *test_cfg*=None, *pretrained*=None, *loss_pose*=None)

Associative embedding pose detectors.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **keypoint_head** (*dict*) – Keypoint head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.
- **loss_pose** (*None*) – Deprecated arguments. Please use `loss_keypoint` for heads instead.

forward(*img*=None, *targets*=None, *masks*=None, *joints*=None, *img metas*=None, *return_loss*=True, *return_heatmap*=False, ***kwargs*)

Calls either `forward_train` or `forward_test` depending on whether `return_loss` is True.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `num_img_channel`: C
 - `img_width`: imgW
 - `img_height`: imgH
 - `heatmaps weight`: W
 - `heatmaps height`: H
 - `max_num_people`: M
-

Parameters

- **img** (*torch.Tensor*[N, C, imgH, imgW]) – Input image.
- **targets** (*list*(*torch.Tensor*[N, K, H, W])) – Multi-scale target heatmaps.
- **masks** (*list*(*torch.Tensor*[N, H, W])) – Masks of multi-scale target heatmaps
- **joints** (*list*(*torch.Tensor*[N, M, K, 2])) – Joints of multi-scale target heatmaps for ae loss
- **img metas** (*dict*) – Information about val & test. By default it includes:
 - `"image_file"`: image path
 - `"aug_data"`: input
 - `"test_scale_factor"`: test scale factor
 - `"base_size"`: base size of input
 - `"center"`: center of image

- "scale": scale of image
- "flip_index": flip index of keypoints
- **loss** (*return*) – return_loss=True for training, return_loss=False for validation & test.
- **return_heatmap** (*bool*) – Option to return heatmap.

Returns if 'return_loss' is true, then return losses. Otherwise, return predicted poses, scores, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(*img*)

Used for computing network FLOPs.

See tools/get_flops.py.

Parameters **img** (*torch.Tensor*) – Input image.

Returns Outputs.

Return type Tensor

forward_test(*img, img metas, return_heatmap=False, **kwargs*)

Inference the bottom-up model.

Note:

- Batchsize: N (currently support batchsize = 1)
 - num_img_channel: C
 - img_width: imgW
 - img_height: imgH
-

Parameters

- **flip_index** (*List(int)*) –
- **aug_data** (*List(Tensor[NxCximgHximgW])*) – Multi-scale image
- **test_scale_factor** (*List(float)*) – Multi-scale factor
- **base_size** (*Tuple(int)*) – Base size of image when scale is 1
- **center** (*np.ndarray*) – center of image
- **scale** (*np.ndarray*) – the scale of image

forward_train(*img, targets, masks, joints, img metas, **kwargs*)

Forward the bottom-up model and calculate the loss.

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
heatmaps weight: W heatmaps height: H max_num_people: M

Parameters

- **img** (*torch.Tensor[N, C, imgH, imgW]*) – Input image.

- **targets** (*List(torch.Tensor[N, K, H, W])*) – Multi-scale target heatmaps.
- **masks** (*List(torch.Tensor[N, H, W])*) – Masks of multi-scale target heatmaps
- **joints** (*List(torch.Tensor[N, M, K, 2])*) – Joints of multi-scale target heatmaps for ae loss
- **img metas** (*dict*) – Information about val&test By default this includes: - “image_file”: image path - “aug_data”: input - “test_scale_factor”: test scale factor - “base_size”: base size of input - “center”: center of image - “scale”: scale of image - “flip_index”: flip index of keypoints

Returns The total loss for bottom-up

Return type dict

init_weights(*pretrained=None*)

Weight initialization for model.

show_result(*img, result, skeleton=None, kpt_score_thr=0.3, bbox_color=None, pose_kpt_color=None, pose_link_color=None, radius=4, thickness=1, font_scale=0.5, win_name="", show=False, show_keypoint_weight=False, wait_time=0, out_file=None*)

Draw *result* over *img*.

Parameters

- **img** (*str or Tensor*) – The image to be displayed.
- **result** (*list[dict]*) – The results to draw over *img* (bbox_result, pose_result).
- **skeleton** (*list[list]*) – The connection of keypoints. skeleton is 0-based indexing.
- **kpt_score_thr** (*float, optional*) – Minimum score of keypoints to be shown. Default: 0.3.
- **pose_kpt_color** (*np.array[Nx3]*) – Color of N keypoints. If None, do not draw keypoints.
- **pose_link_color** (*np.array[Mx3]*) – Color of M links. If None, do not draw links.
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **font_scale** (*float*) – Font scales of texts.
- **win_name** (*str*) – The window name.
- **show** (*bool*) – Whether to show the image. Default: False.
- **show_keypoint_weight** (*bool*) – Whether to change the transparency using the predicted confidence scores of keypoints.
- **wait_time** (*int*) – Value of waitKey param. Default: 0.
- **out_file** (*str or None*) – The filename to write the image. Default: None.

Returns Visualized image only if not *show* or *out_file*

Return type Tensor

property with_keypoint

Check if has keypoint_head.

```
class mmpose.models.detectors.CID(backbone, keypoint_head=None, train_cfg=None, test_cfg=None,
                                  pretrained=None, loss_pose=None)
```

Contextual Instance Decouple for Multi-Person Pose Estimation.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **keypoint_head** (*dict*) – Keypoint head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.
- **loss_pose** (*None*) – Deprecated arguments. Please use `loss_keypoint` for heads instead.

```
forward(img=None, multi_heatmap=None, multi_mask=None, instance_coord=None,
        instance_heatmap=None, instance_mask=None, instance_valid=None, img metas=None,
        return_loss=True, return_heatmap=False, **kwargs)
```

Calls either `forward_train` or `forward_test` depending on whether `return_loss` is `True`.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `num_img_channel`: C
 - `img_width`: imgW
 - `img_height`: imgH
 - `heatmaps weight`: W
 - `heatmaps height`: H
 - `max_num_people`: M
-

Parameters

- **img** (*torch.Tensor*[N, C, imgH, imgW]) – Input image.
- **multi_heatmap** (*torch.Tensor*[N, C, H, W]) – Multi-person heatmaps
- **multi_mask** (*torch.Tensor*[N, 1, H, W]) – Multi-person heatmap mask
- **instance_coord** (*torch.Tensor*[N, M, 2]) – Instance center coord
- **instance_heatmap** (*torch.Tensor*[N, M, C, H, W]) – Single person heatmap for each instance
- **instance_mask** (*torch.Tensor*[N, M, C, 1, 1]) – Single person heatmap mask
- **instance_valid** (*torch.Tensor*[N, M]) – Bool mask to indicate the existence of each person
- **img_metas** (*dict*) – Information about val & test. By default it includes:
 - `"image_file"`: image path
 - `"aug_data"`: input
 - `"test_scale_factor"`: test scale factor

- "base_size": base size of input
- "center": center of image
- "scale": scale of image
- "flip_index": flip index of keypoints
- **loss** (*return*) – return_loss=True for training, return_loss=False for validation & test.
- **return_heatmap** (*bool*) – Option to return heatmap.

Returns if 'return_loss' is true, then return losses. Otherwise, return predicted poses, scores, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(*img*)

Used for computing network FLOPs.

See tools/get_flops.py.

Parameters **img** (*torch.Tensor*) – Input image.

Returns Outputs.

Return type Tensor

forward_test(*img, img metas, return_heatmap=False, **kwargs*)

Inference the bottom-up model.

Note:

- Batchsize: N (currently support batchsize = 1)
 - num_img_channel: C
 - img_width: imgW
 - img_height: imgH
-

Parameters

- **flip_index** (*List(int)*) –
- **aug_data** (*List(Tensor[NxCximgHximgW])*) – Multi-scale image
- **test_scale_factor** (*List(float)*) – Multi-scale factor
- **base_size** (*Tuple(int)*) – Base size of image when scale is 1
- **center** (*np.ndarray*) – center of image
- **scale** (*np.ndarray*) – the scale of image

forward_train(*img, multi_heatmap, multi_mask, instance_coord, instance_heatmap, instance_mask, instance_valid, img metas, **kwargs*)

Forward CID model and calculate the loss.

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH heatmaps weight: W heatmaps height: H max_num_people: M

Parameters

- **img** (*torch.Tensor*[*N, C, imgH, imgW*]) – Input image.
- **multi_heatmap** (*torch.Tensor*[*N, C, H, W*]) – Multi-person heatmaps
- **multi_mask** (*torch.Tensor*[*N, 1, H, W*]) – Multi-person heatmap mask
- **instance_coord** (*torch.Tensor*[*N, M, 2*]) – Instance center coord
- **instance_heatmap** (*torch.Tensor*[*N, M, C, H, W*]) – Single person heatmap for each instance
- **instance_mask** (*torch.Tensor*[*N, M, C, 1, 1*]) – Single person heatmap mask
- **instance_valid** (*torch.Tensor*[*N, M*]) – Bool mask to indicate the existence of each person
- **img metas** (*dict*) – Information about val&test By default this includes: - “image_file”: image path - “aug_data”: input - “test_scale_factor”: test scale factor - “base_size”: base size of input - “center”: center of image - “scale”: scale of image - “flip_index”: flip index of keypoints

Returns The total loss for bottom-up

Return type dict

init_weights(*pretrained=None*)

Weight initialization for model.

show_result(*img, result, skeleton=None, kpt_score_thr=0.3, bbox_color=None, pose_kpt_color=None, pose_link_color=None, radius=4, thickness=1, font_scale=0.5, win_name="", show=False, show_keypoint_weight=False, wait_time=0, out_file=None*)

Draw result over *img*.

Parameters

- **img** (*str* or *Tensor*) – The image to be displayed.
- **result** (*list[dict]*) – The results to draw over *img* (bbox_result, pose_result).
- **skeleton** (*list[list]*) – The connection of keypoints. skeleton is 0-based indexing.
- **kpt_score_thr** (*float, optional*) – Minimum score of keypoints to be shown. Default: 0.3.
- **pose_kpt_color** (*np.array[Nx3]*) – Color of N keypoints. If None, do not draw keypoints.
- **pose_link_color** (*np.array[Mx3]*) – Color of M links. If None, do not draw links.
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **font_scale** (*float*) – Font scales of texts.
- **win_name** (*str*) – The window name.
- **show** (*bool*) – Whether to show the image. Default: False.
- **show_keypoint_weight** (*bool*) – Whether to change the transparency using the predicted confidence scores of keypoints.
- **wait_time** (*int*) – Value of waitKey param. Default: 0.
- **out_file** (*str* or *None*) – The filename to write the image. Default: None.

Returns Visualized image only if not *show* or *out_file*

Return type Tensor

property with_keypoint

Check if has keypoint_head.

```
class mmpose.models.detectors.DetectAndRegress(backbone, human_detector, pose_regressor,
                                              train_cfg=None, test_cfg=None, pretrained=None,
                                              freeze_2d=True)
```

DetectAndRegress approach for multiview human pose detection.

Parameters

- **backbone** (*ConfigDict*) – Dictionary to construct the 2D pose detector
- **human_detector** (*ConfigDict*) – dictionary to construct human detector
- **pose_regressor** (*ConfigDict*) – dictionary to construct pose regressor
- **train_cfg** (*ConfigDict*) – Config for training. Default: None.
- **test_cfg** (*ConfigDict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained 2D model. Default: None.
- **freeze_2d** (*bool*) – Whether to freeze the 2D model in training. Default: True.

```
forward(img=None, img metas=None, return_loss=True, targets=None, masks=None, targets_3d=None,
        input_heatmaps=None, **kwargs)
```

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
feature_maps width: W feature_maps height: H volume_length: cubeL volume_width: cubeW volume_height: cubeH

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **return_loss** – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.
- **targets** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera target feature_maps of the 2D model.
- **masks** (*list(torch.Tensor[NxHxW])*) – Multi-camera masks of the input to the 2D model.
- **targets_3d** (*torch.Tensor[NxcubeLxcubeWxcubeH]*) – Ground-truth 3D heatmap of human centers.
- **input_heatmaps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps when the 2D model is not available. Default: None.
- ****kwargs** –

Returns

if 'return_loss' is true, then return losses. Otherwise, return predicted poses, human centers and sample_id

Return type dict

forward_dummy(img, input_heatmaps=None, num_candidates=5)

Used for computing network FLOPs.

forward_test(img, img metas, input_heatmaps=None)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH feature_maps width: W feature_maps height: H volume_length: cubeL volume_width: cubeW volume_height: cubeH

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **input_heatmaps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps when the 2D model is not available. Default: None.

Returns predicted poses, human centers and sample_id

Return type dict

forward_train(img, img_metas, targets=None, masks=None, targets_3d=None, input_heatmaps=None)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH feature_maps width: W feature_maps height: H volume_length: cubeL volume_width: cubeW volume_height: cubeH

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **targets** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera target feature_maps of the 2D model.
- **masks** (*list(torch.Tensor[NxHxW])*) – Multi-camera masks of the input to the 2D model.
- **targets_3d** (*torch.Tensor[NxcubeLxcubeWxcubeH]*) – Ground-truth 3D heatmap of human centers.
- **input_heatmaps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps when the 2D model is not available. Default: None.

Returns losses.

Return type dict

show_result(*img, img metas, visualize_2d=False, input_heatmaps=None, dataset_info=None, radius=4, thickness=2, out_dir=None, show=False*)

Visualize the results.

train(*mode=True*)

Sets the module in training mode. :param mode: whether to set training mode (True) or evaluation mode (False). Default: True.

Returns self

Return type Module

train_step(*data_batch, optimizer, **kwargs*)

The iteration step during training.

This method defines an iteration step during training, except for the back propagation and optimizer updating, which are done in an optimizer hook. Note that in some complicated cases or models, the whole process including back propagation and optimizer updating is also defined in this method, such as GAN.

Parameters

- **data_batch** (*dict*) – The output of dataloader.
- **optimizer** (*torch.optim.Optimizer | dict*) – The optimizer of runner is passed to `train_step()`. This argument is unused and reserved.

Returns

It should contain at least 3 keys: loss, log_vars, num_samples. `loss` is a tensor for back propagation, which can be a weighted sum of multiple losses. `log_vars` contains all the variables to be sent to the logger. `num_samples` indicates the batch size (when the model is DDP, it means the batch size on each GPU), which is used for averaging the logs.

Return type dict

class mmpose.models.detectors.**DisentangledKeypointRegressor**(*backbone, keypoint_head=None, train_cfg=None, test_cfg=None, pretrained=None*)

Disentangled keypoint regression pose detector.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **keypoint_head** (*dict*) – Keypoint head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.

forward(*img=None, heatmaps=None, masks=None, offsets=None, offset_weights=None, img_metas=None, return_loss=True, return_heatmap=False, **kwargs*)

Calls either `forward_train` or `forward_test` depending on whether `return_loss` is True.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `num_img_channel`: C
 - `img_width`: imgW
 - `img_height`: imgH
 - `heatmaps weight`: W
 - `heatmaps height`: H
 - `max_num_people`: M
-

Parameters

- **img** (*torch.Tensor*[N, C, imgH, imgW]) – # input image.
- **targets** (*list*(*torch.Tensor*[N, K, H, W])) – Multi-scale target heatmaps.
- **masks** (*list*(*torch.Tensor*[N, H, W])) – Masks of multi-scale target heatmaps
- **joints** (*list*(*torch.Tensor*[N, M, K, 2])) – Joints of multi-scale target heatmaps for ae loss
- **img metas** (*dict*) – Information about val & test. By default it includes:
 - “image_file”: image path
 - “aug_data”: # input
 - “test_scale_factor”: test scale factor
 - “base_size”: base size of # input
 - “center”: center of image
 - “scale”: scale of image
 - “flip_index”: flip index of keypoints
- **loss** (*return*) – `return_loss=True` for training, `return_loss=False` for validation & test.
- **return_heatmap** (*bool*) – Option to return heatmap.

Returns if ‘return_loss’ is true, then return losses. Otherwise, return predicted poses, scores, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(img)

Used for computing network FLOPs.

See `tools/get_flops.py`.

Parameters **img** (*torch.Tensor*) – # input image.

Returns Outputs.

Return type Tensor

forward_test(*img, img metas, return_heatmap=False, **kwargs*)

Inference the one-stage model.

Note:

- Batchsize: N (currently support batchsize = 1)
 - num_img_channel: C
 - img_width: imgW
 - img_height: imgH
-

Parameters

- **flip_index** (*List(int)*) –
- **aug_data** (*List(Tensor[NxCximgHximgW])*) – Multi-scale image
- **num_joints** (*int*) – Number of joints of an instance. **test_scale_factor** (*List(float)*): Multi-scale factor
- **base_size** (*Tuple(int)*) – Base size of image when scale is 1
- **image_size** (*int*) – Short edge of images when scale is 1
- **heatmap_size** (*int*) – Short edge of outputs when scale is 1
- **center** (*np.ndarray*) – center of image
- **scale** (*np.ndarray*) – the scale of image
- **skeleton** (*List(List(int))*) – Links of joints

forward_train(*img, heatmaps, masks, offsets, offset_weights, img metas, **kwargs*)

Forward the bottom-up model and calculate the loss.

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
heatmaps weight: W heatmaps height: H max_num_people: M

Parameters

- **img** (*torch.Tensor[N, C, imgH, imgW]*) – # input image.
- **targets** (*List(torch.Tensor[N, K, H, W])*) – Multi-scale target heatmaps.
- **masks** (*List(torch.Tensor[N, H, W])*) – Masks of multi-scale target heatmaps
- **joints** (*List(torch.Tensor[N, M, K, 2])*) – Joints of multi-scale target heatmaps for ae loss
- **img_metas** (*dict*) – Information about val&test By default this includes: - “image_file”: image path - “aug_data”: # input - “test_scale_factor”: test scale factor - “base_size”: base size of # input - “center”: center of image - “scale”: scale of image - “flip_index”: flip index of keypoints

Returns The total loss for bottom-up

Return type dict

init_weights(*pretrained=None*)

Weight initialization for model.

show_result(*img, result, skeleton=None, kpt_score_thr=0.3, bbox_color=None, pose_kpt_color=None, pose_link_color=None, radius=4, thickness=1, font_scale=0.5, win_name="", show=False, show_keypoint_weight=False, wait_time=0, out_file=None*)

Draw *result* over *img*.

Parameters

- **img** (*str* or *Tensor*) – The image to be displayed.
- **result** (*list[dict]*) – The results to draw over *img* (*bbox_result, pose_result*).
- **skeleton** (*list[list]*) – The connection of keypoints. *skeleton* is 0-based indexing.
- **kpt_score_thr** (*float, optional*) – Minimum score of keypoints to be shown. Default: 0.3.
- **pose_kpt_color** (*np.array[Nx3]*) – Color of *N* keypoints. If *None*, do not draw keypoints.
- **pose_link_color** (*np.array[Mx3]*) – Color of *M* links. If *None*, do not draw links.
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **font_scale** (*float*) – Font scales of texts.
- **win_name** (*str*) – The window name.
- **show** (*bool*) – Whether to show the image. Default: *False*.
- **show_keypoint_weight** (*bool*) – Whether to change the transparency using the predicted confidence scores of keypoints.
- **wait_time** (*int*) – Value of *waitKey* param. Default: 0.
- **out_file** (*str* or *None*) – The filename to write the image. Default: *None*.

Returns Visualized image only if not *show* or *out_file*

Return type *Tensor*

property with_keypoint

Check if has keypoint_head.

class `mmpose.models.detectors.GestureRecognizer`(*backbone, neck=None, cls_head=None, train_cfg=None, test_cfg=None, modality='rgb', pretrained=None*)

Hand gesture recognizer.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **neck** (*dict*) – Neck Modules to process feature.
- **cls_head** (*dict*) – Classification head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: *None*.
- **test_cfg** (*dict*) – Config for testing. Default: *None*.
- **modality** (*str* or *list* or *tuple*) – Data modality. Default: *None*.
- **pretrained** (*str*) – Path to the pretrained models.

forward(*video*, *label=None*, *img metas=None*, *return_loss=True*, ***kwargs*)

Calls either `forward_train` or `forward_test` depending on whether `return_loss=True`. Note this setting will change the expected inputs.

Note:

- `batch_size`: N
- `num_vid_channel`: C (Default: 3)
- `video height`: vidH
- `video width`: vidW
- `video length`: vidL

Args: *video* (list[torch.Tensor[NxCxvidLxvidHxvidW]]): Input videos. *label* (torch.Tensor[N]): Category label of videos. *img metas* (list(dict)): Information about data.

By default this includes: - “fps”: video frame rate - “modality”: modality of input videos

return_loss (bool): Option to return loss. *return_loss=True* for training, *return_loss=False* for validation & test.

Returns: dict|tuple: if *return_loss* is true, then return losses. Otherwise, return predicted gestures for clips with a certain length. .

forward_test(*video*, *label*, *img metas*, ***kwargs*)

Defines the computation performed at every call when testing.

forward_train(*video*, *label*, *img metas*, ***kwargs*)

Defines the computation performed at every call when training.

init_weights(*pretrained=None*)

Weight initialization for model.

set_train_epoch(*epoch: int*)

set the training epoch of heads to support customized behaviour.

show_result(*video*, *result*, ***kwargs*)

Visualize the results.

class `mmpose.models.detectors.Interhand3D`(*backbone*, *neck=None*, *keypoint_head=None*,
train_cfg=None, *test_cfg=None*, *pretrained=None*,
loss_pose=None)

Top-down interhand 3D pose detector of paper ref: Gyeongsik Moon.

“InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image”.
A child class of TopDown detector.

forward(*img*, *target=None*, *target_weight=None*, *img metas=None*, *return_loss=True*, ***kwargs*)

Calls either `forward_train` or `forward_test` depending on whether `return_loss=True`. Note this setting will change the expected inputs. When *return_loss=True*, *img* and *img meta* are single-nested (i.e. Tensor and List[dict]), and when *return_loss=False*, *img* and *img meta* should be double nested (i.e. list[Tensor], list[list[dict]]), with the outer list indicating test time augmentations.

Note:

- `batch_size`: N
- `num_keypoints`: K
- `num_img_channel`: C (Default: 3)

- img height: imgH
 - img width: imgW
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **img** (*torch.Tensor[NxCximgHximgW]*) – Input images.
- **target** (*list[torch.Tensor]*) – Target heatmaps, relative hand
- **depth and hand type.** (*root*) –
- **target_weight** (*list[torch.Tensor]*) – Weights for target
- **heatmaps** –
- **hand root depth and hand type.** (*relative*) –
- **img metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - "image_file": path to the image file
 - "center": center of the bbox
 - "scale": scale of the bbox
 - "rotation": rotation of the bbox
 - "bbox_score": score of bbox
 - "heatmap3d_depth_bound": depth bound of hand keypoint 3D heatmap
 - "root_depth_bound": depth bound of relative root depth 1D heatmap
- **return_loss** (*bool*) – Option to return loss. return loss=True for training, return loss=False for validation & test.

Returns if return loss is true, then return losses. Otherwise, return predicted poses, boxes, image paths, heatmaps, relative hand root depth and hand type.

Return type dict[tuple]

forward_test(*img, img_metas, **kwargs*)

Defines the computation performed at every call when testing.

show_result(*result, img=None, skeleton=None, kpt_score_thr=0.3, radius=8, bbox_color='green', thickness=2, pose_kpt_color=None, pose_link_color=None, vis_height=400, num_instances=-1, axis_azimuth=-115, win_name="", show=False, wait_time=0, out_file=None*)

Visualize 3D pose estimation results.

Parameters

- **result** (*list[dict]*) – The pose estimation results containing:
 - "keypoints_3d" ([K,4]): 3D keypoints
 - "keypoints" ([K,3] or [T,K,3]): **Optional for visualizing** 2D inputs. If a sequence is given, only the last frame will be used for visualization
 - "bbox" ([4,] or [T,4]): **Optional for visualizing** 2D inputs

- “title” (str): title for the subplot
- **img** (str or Tensor) – Optional. The image to visualize 2D inputs on.
- **skeleton** (list of [idx_i, idx_j]) – Skeleton described by a list of links, each is a pair of joint indices.
- **kpt_score_thr** (float, optional) – Minimum score of keypoints to be shown. Default: 0.3.
- **radius** (int) – Radius of circles.
- **bbox_color** (str or tuple or Color) – Color of bbox lines.
- **thickness** (int) – Thickness of lines.
- **pose_kpt_color** (np.array[Nx3]) – Color of N keypoints. If None, do not draw keypoints.
- **pose_link_color** (np.array[Mx3]) – Color of M limbs. If None, do not draw limbs.
- **vis_height** (int) – The image height of the visualization. The width will be N*vis_height depending on the number of visualized items.
- **num_instances** (int) – Number of instances to be shown in 3D. If smaller than 0, all the instances in the pose_result will be shown. Otherwise, pad or truncate the pose_result to a length of num_instances.
- **axis_azimuth** (float) – axis azimuth angle for 3D visualizations.
- **win_name** (str) – The window name.
- **show** (bool) – Whether to show the image. Default: False.
- **wait_time** (int) – Value of waitKey param. Default: 0.
- **out_file** (str or None) – The filename to write the image. Default: None.

Returns Visualized img, only if not *show* or *out_file*.

Return type Tensor

```
class mmpose.models.detectors.MultiTask(backbone, heads, necks=None, head2neck=None,
                                         pretrained=None)
```

Multi-task detectors.

Parameters

- **backbone** (dict) – Backbone modules to extract feature.
- **heads** (list[dict]) – heads to output predictions.
- **necks** (list[dict] | None) – necks to process feature.
- **(dict{int (head2neck) – int})**: head index to neck index.
- **pretrained** (str) – Path to the pretrained models.

forward(img, target=None, target_weight=None, img metas=None, return_loss=True, **kwargs)

Calls either forward_train or forward_test depending on whether return_loss=True. Note this setting will change the expected inputs. When *return_loss=True*, img and img_meta are single-nested (i.e. Tensor and List[dict]), and when *return_loss=False*, img and img_meta should be double nested (i.e. List[Tensor], List[List[dict]]), with the outer list indicating test time augmentations.

Note:

- batch_size: N
 - num_keypoints: K
 - num_img_channel: C (Default: 3)
 - img height: imgH
 - img weight: imgW
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **img** (*torch.Tensor*[N,C,imgH,imgW]) – Input images.
- **target** (*list*[*torch.Tensor*]) – Targets.
- **target_weight** (*List*[*torch.Tensor*]) – Weights.
- **img metas** (*list*(*dict*)) – Information about data augmentation By default this includes:
 - "image_file": path to the image file
 - "center": center of the bbox
 - "scale": scale of the bbox
 - "rotation": rotation of the bbox
 - "bbox_score": score of bbox
- **return_loss** (*bool*) – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.

Returns if *return loss* is true, then return losses. Otherwise, return predicted poses, boxes, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(img)

Used for computing network FLOPs.

See tools/get_flops.py.

Parameters **img** (*torch.Tensor*) – Input image.

Returns Outputs.

Return type list[*Tensor*]

forward_test(img, img_metas, **kwargs)

Defines the computation performed at every call when testing.

forward_train(img, target, target_weight, img_metas, **kwargs)

Defines the computation performed at every call when training.

init_weights(pretrained=None)

Weight initialization for model.

property with_necks

Check if has keypoint_head.

```
class mmpose.models.detectors.ParametricMesh(backbone, mesh_head, smpl, disc=None, loss_gan=None,
                                              loss_mesh=None, train_cfg=None, test_cfg=None,
                                              pretrained=None)
```

Model-based 3D human mesh detector. Take a single color image as input and output 3D joints, SMPL parameters and camera parameters.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **mesh_head** (*dict*) – Mesh head to process feature.
- **smpl** (*dict*) – Config for SMPL model.
- **disc** (*dict*) – Discriminator for SMPL parameters. Default: None.
- **loss_gan** (*dict*) – Config for adversarial loss. Default: None.
- **loss_mesh** (*dict*) – Config for mesh loss. Default: None.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.

```
forward(img, img metas=None, return_loss=False, **kwargs)
```

Forward function.

Calls either forward_train or forward_test depending on whether return_loss=True.

Note:

- batch_size: N
 - num_img_channel: C (Default: 3)
 - img height: imgH
 - img width: imgW
-

Parameters

- **img** (*torch.Tensor[N x C x imgH x imgW]*) – Input images.
- **img_metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - "image_file": path to the image file
 - "center": center of the bbox
 - "scale": scale of the bbox
 - "rotation": rotation of the bbox
 - "bbox_score": score of bbox
- **return_loss** (*bool*) – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.

Returns Return predicted 3D joints, SMPL parameters, boxes and image paths.

forward_dummy(*img*)

Used for computing network FLOPs.

See `tools/get_flops.py`.

Parameters **img** (*torch.Tensor*) – Input image.

Returns Outputs.

Return type Tensor

forward_test(*img, img metas, return_vertices=False, return_faces=False, **kwargs*)

Defines the computation performed at every call when testing.

forward_train(*args, **kwargs)

Forward function for training.

For ParametricMesh, we do not use this interface.

get_3d_joints_from_mesh(*vertices*)

Get 3D joints from 3D mesh using predefined joints regressor.

init_weights(*pretrained=None*)

Weight initialization for model.

show_result(*result, img, show=False, out_file=None, win_name="", wait_time=0, bbox_color='green', mesh_color=(76, 76, 204), **kwargs*)

Visualize 3D mesh estimation results.

Parameters

- **result** (*list[dict]*) – The mesh estimation results containing:
 - “bbox” (*ndarray[4]*): instance bounding bbox
 - “center” (*ndarray[2]*): bbox center
 - “scale” (*ndarray[2]*): bbox scale
 - “keypoints_3d” (*ndarray[K,3]*): predicted 3D keypoints
 - “camera” (*ndarray[3]*): camera parameters
 - “vertices” (*ndarray[V, 3]*): predicted 3D vertices
 - “faces” (*ndarray[F, 3]*): mesh faces
- **img** (*str or Tensor*) – Optional. The image to visualize 2D inputs on.
- **win_name** (*str*) – The window name.
- **show** (*bool*) – Whether to show the image. Default: False.
- **wait_time** (*int*) – Value of waitKey param. Default: 0.
- **out_file** (*str or None*) – The filename to write the image. Default: None.
- **bbox_color** (*str or tuple or Color*) – Color of bbox lines.
- **mesh_color** (*str or tuple or Color*) – Color of mesh surface.

Returns Visualized img, only if not *show* or *out_file*.

Return type ndarray

train_step(*data_batch, optimizer, **kwargs*)

Train step function.

In this function, the detector will finish the train step following the pipeline:

1. get fake and real SMPL parameters
2. optimize discriminator (if have)
3. optimize generator

If `self.train_cfg.disc_step > 1`, the train step will contain multiple iterations for optimizing discriminator with different input data and only one iteration for optimizing generator after `disc_step` iterations for discriminator.

Parameters

- **data_batch** (`torch.Tensor`) – Batch of data as input.
- **optimizer** (`dict[torch.optim.Optimizer]`) – Dict with optimizers for generator and discriminator (if have).

Returns Dict with loss, information for logger, the number of samples.

Return type outputs (dict)

val_step(`data_batch`, `**kwargs`)

Forward function for evaluation.

Parameters **data_batch** (`dict`) – Contain data for forward.

Returns Contain the results from model.

Return type dict

```
class mmpose.models.detectors.PoseLifter(backbone, neck=None, keypoint_head=None,
                                         traj_backbone=None, traj_neck=None, traj_head=None,
                                         loss_semi=None, train_cfg=None, test_cfg=None,
                                         pretrained=None)
```

Pose lifter that lifts 2D pose to 3D pose.

The basic model is a pose model that predicts root-relative pose. If `traj_head` is not `None`, a trajectory model that predicts absolute root joint position is also built.

Parameters

- **backbone** (`dict`) – Config for the backbone of pose model.
- **neck** (`dict/None`) – Config for the neck of pose model.
- **keypoint_head** (`dict/None`) – Config for the head of pose model.
- **traj_backbone** (`dict/None`) – Config for the backbone of trajectory model. If `traj_backbone` is `None` and `traj_head` is not `None`, trajectory model will share backbone with pose model.
- **traj_neck** (`dict/None`) – Config for the neck of trajectory model.
- **traj_head** (`dict/None`) – Config for the head of trajectory model.
- **loss_semi** (`dict/None`) – Config for semi-supervision loss.
- **train_cfg** (`dict/None`) – Config for keypoint head during training.
- **test_cfg** (`dict/None`) – Config for keypoint head during testing.
- **pretrained** (`str/None`) – Path to pretrained weights.

forward(`input`, `target=None`, `target_weight=None`, `metas=None`, `return_loss=True`, `**kwargs`)

Calls either `forward_train` or `forward_test` depending on whether `return_loss=True`.

Note:

- `batch_size`: N
 - `num_input_keypoints`: Ki
 - `input_keypoint_dim`: Ci
 - `input_sequence_len`: Ti
 - `num_output_keypoints`: Ko
 - `output_keypoint_dim`: Co
 - `input_sequence_len`: To
-

Parameters

- **input** (*torch.Tensor*[NxKixCixTi]) – Input keypoint coordinates.
- **target** (*torch.Tensor*[NxKoxCoxTo]) – Output keypoint coordinates. Defaults to None.
- **target_weight** (*torch.Tensor*[NxKox1]) – Weights across different joint types. Defaults to None.
- **metas** (*list(dict)*) – Information about data augmentation
- **return_loss** (*bool*) – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.

Returns If *reutrnr_loss* is true, return losses. Otherwise return predicted poses.

Return type dict|Tensor

forward_dummy(input)

Used for computing network FLOPs. See `tools/get_flops.py`.

Parameters **input** (*torch.Tensor*) – Input pose

Returns Model output

Return type Tensor

forward_test(input, metas, **kwargs)

Defines the computation performed at every call when training.

forward_train(input, target, target_weight, metas, **kwargs)

Defines the computation performed at every call when training.

init_weights(pretrained=None)

Weight initialization for model.

show_result(result, img=None, skeleton=None, pose_kpt_color=None, pose_link_color=None, radius=8, thickness=2, vis_height=400, num_instances=-1, axis_azimuth=70, win_name="", show=False, wait_time=0, out_file=None)

Visualize 3D pose estimation results.

Parameters

- **result** (*list[dict]*) – The pose estimation results containing:
 - `"keypoints_3d"` ([K,4]): 3D keypoints

- “keypoints” ([K,3] or [T,K,3]): **Optional for visualizing** 2D inputs. If a sequence is given, only the last frame will be used for visualization
- “bbox” ([4,] or [T,4]): Optional for visualizing 2D inputs
- “title” (str): title for the subplot
- **img** (str or Tensor) – Optional. The image to visualize 2D inputs on.
- **skeleton** (list of [idx_i, idx_j]) – Skeleton described by a list of links, each is a pair of joint indices.
- **pose_kpt_color** (np.array[Nx3]) – Color of N keypoints. If None, do not draw keypoints.
- **pose_link_color** (np.array[Mx3]) – Color of M links. If None, do not draw links.
- **radius** (int) – Radius of circles.
- **thickness** (int) – Thickness of lines.
- **vis_height** (int) – The image height of the visualization. The width will be N*vis_height depending on the number of visualized items.
- **num_instances** (int) – Number of instances to be shown in 3D. If smaller than 0, all the instances in the result will be shown. Otherwise, pad or truncate the result to a length of num_instances.
- **axis_azimuth** (float) – axis azimuth angle for 3D visualizations.
- **win_name** (str) – The window name.
- **show** (bool) – Whether to directly show the visualization.
- **wait_time** (int) – Value of waitKey param. Default: 0.
- **out_file** (str or None) – The filename to write the image. Default: None.

Returns Visualized img, only if not *show* or *out_file*.

Return type Tensor

property with_keypoint

Check if has keypoint_head.

property with_neck

Check if has keypoint_neck.

property with_traj

Check if has trajectory_head.

property with_traj_backbone

Check if has trajectory_backbone.

property with_traj_neck

Check if has trajectory_neck.

class mmpose.models.detectors.**PoseWarper**(backbone, neck=None, keypoint_head=None, train_cfg=None, test_cfg=None, pretrained=None, loss_pose=None, concat_tensors=True)

Top-down pose detectors for multi-frame settings for video inputs.

“Learning temporal pose estimation from sparsely-labeled videos”.

A child class of TopDown detector. The main difference between PoseWarper and TopDown lies in that the former takes a list of tensors as input image while the latter takes a single tensor as input image in forward method.

Parameters

- **backbone** (*dict*) – Backbone modules to extract features.
- **neck** (*dict*) – intermediate modules to transform features.
- **keypoint_head** (*dict*) – Keypoint head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.
- **loss_pose** (*None*) – Deprecated arguments. Please use *loss_keypoint* for heads instead.
- **concat_tensors** (*bool*) – Whether to concat the tensors on the batch dim, which can speed up, Default: True

forward(*img*, *target=None*, *target_weight=None*, *img metas=None*, *return_loss=True*, *return_heatmap=False*, ***kwargs*)

Calls either forward_train or forward_test depending on whether return_loss=True. Note this setting will change the expected inputs. When *return_loss=True*, *img* and *img_meta* are single-nested (i.e. Tensor and List[dict]), and when *return_loss=False*, *img* and *img_meta* should be double nested (i.e. List[Tensor], List[List[dict]]), with the outer list indicating test time augmentations.

Note:

- number of frames: F
 - batch_size: N
 - num_keypoints: K
 - num_img_channel: C (Default: 3)
 - img height: imgH
 - img width: imgW
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **imgs** (*list[F, torch.Tensor[N, C, imgH, imgW]]*) – multiple input frames
- **target** (*torch.Tensor[N, K, H, W]*) – Target heatmaps for one frame.
- **target_weight** (*torch.Tensor[N, K, 1]*) – Weights across different joint types.
- **img_metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - "image_file": paths to multiple video frames
 - "center": center of the bbox
 - "scale": scale of the bbox

- "rotation": rotation of the bbox
- "bbox_score": score of bbox
- **return_loss** (*bool*) – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.
- **return_heatmap** (*bool*) – Option to return heatmap.

Returns if *return loss* is true, then return losses. Otherwise, return predicted poses, boxes, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(*img*)

Used for computing network FLOPs.

See tools/get_flops.py.

Parameters **img** (*torch.Tensor[N,C,imgH,imgW]*, or *list/tuple of tensors*) – multiple input frames, $N \geq 2$.

Returns Output heatmaps.

Return type Tensor

forward_test(*imgs, img metas, return_heatmap=False, **kwargs*)

Defines the computation performed at every call when testing.

forward_train(*imgs, target, target_weight, img metas, **kwargs*)

Defines the computation performed at every call when training.

class mmpose.models.detectors.**TopDown**(*backbone, neck=None, keypoint_head=None, train_cfg=None, test_cfg=None, pretrained=None, loss_pose=None*)

Top-down pose detectors.

Parameters

- **backbone** (*dict*) – Backbone modules to extract feature.
- **keypoint_head** (*dict*) – Keypoint head to process feature.
- **train_cfg** (*dict*) – Config for training. Default: None.
- **test_cfg** (*dict*) – Config for testing. Default: None.
- **pretrained** (*str*) – Path to the pretrained models.
- **loss_pose** (*None*) – Deprecated arguments. Please use *loss_keypoint* for heads instead.

forward(*img, target=None, target_weight=None, img metas=None, return_loss=True, return_heatmap=False, **kwargs*)

Calls either *forward_train* or *forward_test* depending on whether *return_loss=True*. Note this setting will change the expected inputs. When *return_loss=True*, *img* and *img meta* are single-nested (i.e. Tensor and List[dict]), and when *return_loss=False*, *img* and *img meta* should be double nested (i.e. List[Tensor], List[List[dict]]), with the outer list indicating test time augmentations.

Note:

- *batch_size*: N
- *num_keypoints*: K
- *num_img_channel*: C (Default: 3)

- img height: imgH
 - img width: imgW
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **img** (*torch.Tensor[NxCximgHximgW]*) – Input images.
- **target** (*torch.Tensor[NxKxHxW]*) – Target heatmaps.
- **target_weight** (*torch.Tensor[NxKx1]*) – Weights across different joint types.
- **img metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - "image_file": path to the image file
 - "center": center of the bbox
 - "scale": scale of the bbox
 - "rotation": rotation of the bbox
 - "bbox_score": score of bbox
- **return_loss** (*bool*) – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.
- **return_heatmap** (*bool*) – Option to return heatmap.

Returns if *return loss* is true, then return losses. Otherwise, return predicted poses, boxes, image paths and heatmaps.

Return type dict[tuple]

forward_dummy(img)

Used for computing network FLOPs.

See tools/get_flops.py.

Parameters **img** (*torch.Tensor*) – Input image.

Returns Output heatmaps.

Return type Tensor

forward_test(img, img_metas, return_heatmap=False, **kwargs)

Defines the computation performed at every call when testing.

forward_train(img, target, target_weight, img_metas, **kwargs)

Defines the computation performed at every call when training.

init_weights(pretrained=None)

Weight initialization for model.

show_result(img, result, skeleton=None, kpt_score_thr=0.3, bbox_color='green', pose_kpt_color=None, pose_link_color=None, text_color='white', radius=4, thickness=1, font_scale=0.5, bbox_thickness=1, win_name="", show=False, show_keypoint_weight=False, wait_time=0, out_file=None)

Draw *result* over *img*.

Parameters

- **img** (*str* or *Tensor*) – The image to be displayed.
- **result** (*list[dict]*) – The results to draw over *img* (bbox_result, pose_result).
- **skeleton** (*list[list]*) – The connection of keypoints. skeleton is 0-based indexing.
- **kpt_score_thr** (*float*, *optional*) – Minimum score of keypoints to be shown. Default: 0.3.
- **bbox_color** (*str* or *tuple* or *Color*) – Color of bbox lines.
- **pose_kpt_color** (*np.array[Nx3]*) – Color of N keypoints. If None, do not draw keypoints.
- **pose_link_color** (*np.array[Mx3]*) – Color of M links. If None, do not draw links.
- **text_color** (*str* or *tuple* or *Color*) – Color of texts.
- **radius** (*int*) – Radius of circles.
- **thickness** (*int*) – Thickness of lines.
- **font_scale** (*float*) – Font scales of texts.
- **win_name** (*str*) – The window name.
- **show** (*bool*) – Whether to show the image. Default: False.
- **show_keypoint_weight** (*bool*) – Whether to change the transparency using the predicted confidence scores of keypoints.
- **wait_time** (*int*) – Value of waitKey param. Default: 0.
- **out_file** (*str* or *None*) – The filename to write the image. Default: None.

Returns Visualized img, only if not *show* or *out_file*.

Return type Tensor

property with_keypoint

Check if has keypoint_head.

property with_neck

Check if has neck.

```
class mmpose.models.detectors.VoxelCenterDetector(image_size, heatmap_size, space_size, cube_size,
                                                    space_center, center_net, center_head,
                                                    train_cfg=None, test_cfg=None)
```

Detect human center by 3D CNN on voxels.

Please refer to the *paper* <<https://arxiv.org/abs/2004.06239>> for details. :param image_size: input size of the 2D model. :type image_size: list :param heatmap_size: output size of the 2D model. :type heatmap_size: list :param space_size: Size of the 3D space. :type space_size: list :param cube_size: Size of the input volume to the 3D CNN. :type cube_size: list :param space_center: Coordinate of the center of the 3D space. :type space_center: list :param center_net: Dictionary to construct the center net. :type center_net: ConfigDict :param center_head: Dictionary to construct the center head. :type center_head: ConfigDict :param train_cfg: Config for training. Default: None. :type train_cfg: ConfigDict :param test_cfg: Config for testing. Default: None. :type test_cfg: ConfigDict

```
assign2gt(center_candidates, gt_centers, gt_num_persons)
```

“Assign gt id to each valid human center candidate.

forward(*img, img metas, return_loss=True, feature_maps=None, targets_3d=None*)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
heatmaps width: W heatmaps height: H

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **return_loss** – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.
- **targets_3d** (*torch.Tensor[NxcubeLxcubeWxcubeH]*) – Ground-truth 3D heatmap of human centers.
- **feature_maps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps.

Returns

if ‘return_loss’ is true, then return losses. Otherwise, return predicted poses

Return type dict

forward_dummy(*feature_maps*)

Used for computing network FLOPs.

forward_test(*img, img metas, feature_maps=None*)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
heatmaps width: W heatmaps height: H

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **feature_maps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps.

Returns human centers

forward_train(*img, img metas, feature_maps=None, targets_3d=None, return_preds=False*)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH
heatmaps width: W heatmaps height: H

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **img metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **targets_3d** (*torch.Tensor[NxcubeLxcubeWxcubeH]*) – Ground-truth 3D heatmap of human centers.
- **feature_maps** (*list(torch.Tensor[NxKxHxW])*) – Multi-camera feature_maps.
- **return_preds** (*bool*) – Whether to return prediction results

Returns

if ‘return_pred’ is true, then return losses and human centers. Otherwise, return losses only

Return type dict

show_result(***kwargs*)
Visualize the results.

class mmpose.models.detectors.**VoxelSinglePose**(*image_size, heatmap_size, sub_space_size, sub_cube_size, num_joints, pose_net, pose_head, train_cfg=None, test_cfg=None*)

VoxelPose Please refer to the *paper* <<https://arxiv.org/abs/2004.06239>> for details.

Parameters

- **image_size** (*list*) – input size of the 2D model.
- **heatmap_size** (*list*) – output size of the 2D model.
- **sub_space_size** (*list*) – Size of the cuboid human proposal.
- **sub_cube_size** (*list*) – Size of the input volume to the pose net.
- **pose_net** (*ConfigDict*) – Dictionary to construct the pose net.
- **pose_head** (*ConfigDict*) – Dictionary to construct the pose head.
- **train_cfg** (*ConfigDict*) – Config for training. Default: None.
- **test_cfg** (*ConfigDict*) – Config for testing. Default: None.

forward(*img, img_metas, return_loss=True, feature_maps=None, human_candidates=None, **kwargs*)

Note: batch_size: N num_keypoints: K num_img_channel: C img_width: imgW img_height: imgH feature_maps width: W feature_maps height: H volume_length: cubeL volume_width: cubeW volume_height: cubeH

Parameters

- **img** (*list(torch.Tensor[NxCximgHximgW])*) – Multi-camera input images to the 2D model.
- **feature_maps** (*list(torch.Tensor[NxCxHxW])*) – Multi-camera input feature_maps.
- **img metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.

- **human_candidates** (*torch.Tensor*[NxPx5]) – Human candidates.
- **return_loss** – Option to *return loss*. *return loss=True* for training, *return loss=False* for validation & test.

forward_dummy(*feature_maps*, *num_candidates=5*)

Used for computing network FLOPs.

forward_test(*img*, *img metas*, *feature_maps=None*, *human_candidates=None*, ***kwargs*)

Defines the computation performed at training. .. note:

```
batch_size: N
num_keypoints: K
num_img_channel: C
img_width: imgW
img_height: imgH
feature_maps width: W
feature_maps height: H
volume_length: cubeL
volume_width: cubeW
volume_height: cubeH
```

Parameters

- **img** (*list(torch.Tensor*[NxCximgHximgW])) – Multi-camera input images to the 2D model.
- **feature_maps** (*list(torch.Tensor*[NxCxHxW])) – Multi-camera input feature_maps.
- **img_metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **human_candidates** (*torch.Tensor*[NxPx5]) – Human candidates.

Returns predicted poses, human centers and sample_id

Return type dict

forward_train(*img*, *img metas*, *feature_maps=None*, *human_candidates=None*, *return_preds=False*, ***kwargs*)

Defines the computation performed at training. .. note:

```
batch_size: N
num_keypoints: K
num_img_channel: C
img_width: imgW
img_height: imgH
feature_maps width: W
feature_maps height: H
volume_length: cubeL
volume_width: cubeW
volume_height: cubeH
```

Parameters

- **img** (*list(torch.Tensor*[NxCximgHximgW])) – Multi-camera input images to the 2D model.

- **feature_maps** (*list(torch.Tensor[NxCxHxW])*) – Multi-camera input feature_maps.
- **img metas** (*list(dict)*) – Information about image, 3D groundtruth and camera parameters.
- **human_candidates** (*torch.Tensor[NxPx5]*) – Human candidates.
- **return_preds** (*bool*) – Whether to return prediction results

Returns losses.

Return type dict

show_result(***kwargs*)
Visualize the results.

49.4 heads

```
class mmpose.models.heads.AEHigherResolutionHead(in_channels, num_joints, tag_per_joint=True,
                                                  extra=None, num_deconv_layers=1,
                                                  num_deconv_filters=(32), num_deconv_kernels=(4),
                                                  num_basic_blocks=4, cat_output=None,
                                                  with_ae_loss=None, loss_keypoint=None)
```

Associative embedding with higher resolution head. paper ref: Bowen Cheng et al. “HigherHRNet: Scale-Aware Representation Learning for Bottom- Up Human Pose Estimation”.

Parameters

- **in_channels** (*int*) – Number of input channels.
- **num_joints** (*int*) – Number of joints
- **tag_per_joint** (*bool*) – If tag_per_joint is True, the dimension of tags equals to num_joints, else the dimension of tags is 1. Default: True
- **extra** (*dict*) – Configs for extra conv layers. Default: None
- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should >= 0. Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers > 0, the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **cat_output** (*list[bool]*) – Option to concat outputs.
- **with_ae_loss** (*list[bool]*) – Option to use ae loss.
- **loss_keypoint** (*dict*) – Config for loss. Default: None.

forward(*x*)
Forward function.

get_loss(*outputs, targets, masks, joints*)
Calculate bottom-up keypoint loss.

Note:

- batch_size: N

- num_keypoints: K
 - num_outputs: O
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **outputs** (*list(torch.Tensor[N,K,H,W])*) – Multi-scale output heatmaps.
- **targets** (*List(torch.Tensor[N,K,H,W])*) – Multi-scale target heatmaps.
- **masks** (*List(torch.Tensor[N,H,W])*) – Masks of multi-scale target heatmaps
- **joints** (*List(torch.Tensor[N,M,K,2])*) – Joints of multi-scale target heatmaps for ae loss

init_weights()

Initialize model weights.

```
class mmpose.models.heads.AEMultiStageHead(in_channels, out_channels, num_stages=1,  
                                           num_deconv_layers=3, num_deconv_filters=(256, 256,  
                                           256), num_deconv_kernels=(4, 4, 4), extra=None,  
                                           loss_keypoint=None)
```

Associative embedding multi-stage head. paper ref: Alejandro Newell et al. “Associative Embedding: End-to-end Learning for Joint Detection and Grouping”

Parameters

- **in_channels** (*int*) – Number of input channels.
- **out_channels** (*int*) – Number of output channels.
- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should >= 0. Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers > 0, the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **loss_keypoint** (*dict*) – Config for loss. Default: None.

forward(x)

Forward function.

Returns a list of heatmaps from multiple stages.

Return type out (list[*Tensor*])

get_loss(output, targets, masks, joints)

Calculate bottom-up keypoint loss.

Note:

- batch_size: N
- num_keypoints: K
- heatmaps height: H
- heatmaps weight: W

Parameters

- **output** (*List(torch.Tensor[NxKxHxW])*) – Output heatmaps.
- **targets** (*List(List(torch.Tensor[NxKxHxW]))*) – Multi-stage and multi-scale target heatmaps.
- **masks** (*List(List(torch.Tensor[NxHxW]))*) – Masks of multi-stage and multi-scale target heatmaps
- **joints** (*List(List(torch.Tensor[NxMxKx2]))*) – Joints of multi-stage multi-scale target heatmaps for ae loss

init_weights()

Initialize model weights.

```
class mmpose.models.heads.AESimpleHead(in_channels, num_joints, num_deconv_layers=3,
                                       num_deconv_filters=(256, 256, 256), num_deconv_kernels=(4, 4,
                                       4), tag_per_joint=True, with_ae_loss=None, extra=None,
                                       loss_keypoint=None)
```

Associative embedding simple head. paper ref: Alejandro Newell et al. “Associative Embedding: End-to-end Learning for Joint Detection and Grouping”

Parameters

- **in_channels** (*int*) – Number of input channels.
- **num_joints** (*int*) – Number of joints.
- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should >= 0. Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers > 0, the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **tag_per_joint** (*bool*) – If tag_per_joint is True, the dimension of tags equals to num_joints, else the dimension of tags is 1. Default: True
- **with_ae_loss** (*list[bool]*) – Option to use ae loss or not.
- **loss_keypoint** (*dict*) – Config for loss. Default: None.

get_loss(outputs, targets, masks, joints)

Calculate bottom-up keypoint loss.

Note:

- batch_size: N
- num_keypoints: K
- num_outputs: O
- heatmaps height: H
- heatmaps weight: W

Parameters

- **outputs** (*list(torch.Tensor[N,K,H,W])*) – Multi-scale output heatmaps.
- **targets** (*List(torch.Tensor[N,K,H,W])*) – Multi-scale target heatmaps.
- **masks** (*List(torch.Tensor[N,H,W])*) – Masks of multi-scale target heatmaps
- **joints** (*List(torch.Tensor[N,M,K,2])*) – Joints of multi-scale target heatmaps for ae loss

```
class mmpose.models.heads.CIDHead(in_channels, gfd_channels, num_joints, multi_hm_loss_factor=1.0,  
                                  single_hm_loss_factor=4.0, contrastive_loss_factor=1.0,  
                                  max_train_instances=200, prior_prob=0.01)
```

CID head. paper ref: Dongkai Wang et al. “Contextual Instance Decouple for Robust Multi-Person Pose Estimation”.

Parameters

- **in_channels** (*int*) – Number of input channels.
- **gfd_channels** (*int*) – Number of instance feature map channels
- **num_joints** (*int*) – Number of joints
- **multi_hm_loss_factor** (*float*) – loss weight for multi-person heatmap
- **single_hm_loss_factor** (*float*) – loss weight for single person heatmap
- **contrastive_loss_factor** (*float*) – loss weight for contrastive loss
- **max_train_instances** (*int*) – limit the number of instances
- **training to avoid** (*during*) –
- **prior_prob** (*float*) – focal loss bias initialization

forward(*features, forward_info=None*)
Forward function.

init_weights()
Initialize model weights.

```
class mmpose.models.heads.CuboidCenterHead(space_size, space_center, cube_size, max_num=10,  
                                            max_pool_kernel=3)
```

Get results from the 3D human center heatmap. In this module, human 3D centers are local maximums obtained from the 3D heatmap via NMS (max- pooling).

Parameters

- **space_size** (*list[3]*) – The size of the 3D space.
- **cube_size** (*list[3]*) – The size of the heatmap volume.
- **space_center** (*list[3]*) – The coordinate of space center.
- **max_num** (*int*) – Maximum of human center detections.
- **max_pool_kernel** (*int*) – Kernel size of the max-pool kernel in nms.

forward(*heatmap_volumes*)

Parameters **heatmap_volumes** (*torch.Tensor(NXLXWXH)*) – 3D human center heatmaps predicted by the network.

Returns Coordinates of human centers.

Return type **human_centers** (*torch.Tensor(NXPX5)*)


```
class mmpose.models.heads.CuboidPoseHead(beta)
```

```
forward(heatmap_volumes, grid_coordinates)
```

Parameters

- **heatmap_volumes** (*torch.Tensor* ($N \times K \times L \times W \times H$)) – 3D human pose heatmaps predicted by the network.
- **grid_coordinates** (*torch.Tensor* ($N \times (L \times W \times H) \times 3$)) – Coordinates of the grids in the heatmap volumes.

Returns Coordinates of human poses.

Return type human_poses (*torch.Tensor* ($N \times K \times 3$))

```
class mmpose.models.heads.DEKRHead(in_channels, num_joints, num_heatmap_filters=32,
                                   num_offset_filters_per_joint=15, in_index=0, input_transform=None,
                                   num_deconv_layers=0, num_deconv_filters=None,
                                   num_deconv_kernels=None, extra={'final_conv_kernel': 0},
                                   align_corners=False, heatmap_loss=None, offset_loss=None)
```

DisEntangled Keypoint Regression head. “Bottom-up human pose estimation via disentangled keypoint regression”, CVPR’2021.

Parameters

- **in_channels** (*int*) – Number of input channels.
- **num_joints** (*int*) – Number of joints.
- **num_heatmap_filters** (*int*) – Number of filters for heatmap branch.
- **num_offset_filters_per_joint** (*int*) – Number of filters for each joint.
- **in_index** (*int* / *Sequence* [*int*]) – Input feature index. Default: 0
- **input_transform** (*str* / *None*) – Transformation type of input features. Options: ‘resize_concat’, ‘multiple_select’, *None*. Default: *None*.
 - **‘resize_concat’**: Multiple feature maps will be resized to the same size as the first one and then concat together. Usually used in FCN head of HRNet.
 - **‘multiple_select’**: Multiple feature maps will be bundle into a list and passed into decode head.
 - *None*: Only one select feature map is allowed.
- **align_corners** (*bool*) – align_corners argument of *F.interpolate*. Default: *False*.
- **heatmap_loss** (*dict*) – Config for heatmap loss. Default: *None*.
- **offset_loss** (*dict*) – Config for offset loss. Default: *None*.

```
forward(x)
```

Forward function.

```
get_loss(outputs, heatmaps, masks, offsets, offset_weights)
```

Calculate the dekr loss.

Note:

- batch_size: N

- `num_channels`: C
 - `num_joints`: K
 - `heatmaps height`: H
 - `heatmaps weight`: W
-

Parameters

- **outputs** (*List(torch.Tensor[N, C, H, W])*) – Multi-scale outputs.
- **heatmaps** (*List(torch.Tensor[N, K+1, H, W])*) – Multi-scale heatmap targets.
- **masks** (*List(torch.Tensor[N, K+1, H, W])*) – Weights of multi-scale heatmap targets.
- **offsets** (*List(torch.Tensor[N, K*2, H, W])*) – Multi-scale offset targets.
- **offset_weights** (*List(torch.Tensor[N, K*2, H, W])*) – Weights of multi-scale offset targets.

`init_weights()`

Initialize model weights.

```
class mmpose.models.heads.DeconvHead(in_channels=3, out_channels=17, num_deconv_layers=3,
                                     num_deconv_filters=(256, 256, 256), num_deconv_kernels=(4, 4,
                                     4), extra=None, in_index=0, input_transform=None,
                                     align_corners=False, loss_keypoint=None)
```

Simple deconv head.

Parameters

- **in_channels** (*int*) – Number of input channels.
- **out_channels** (*int*) – Number of output channels.
- **num_deconv_layers** (*int*) – Number of deconv layers. `num_deconv_layers` should ≥ 0 . Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If `num_deconv_layers` > 0 , the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **in_index** (*int/Sequence[int]*) – Input feature index. Default: 0
- **input_transform** (*str/None*) – Transformation type of input features. Options: ‘resize_concat’, ‘multiple_select’, None. Default: None.
 - ‘resize_concat’: Multiple feature maps will be resized to the same size as the first one and then concat together. Usually used in FCN head of HRNet.
 - ‘multiple_select’: Multiple feature maps will be bundle into a list and passed into decode head.
 - None: Only one select feature map is allowed.
- **align_corners** (*bool*) – `align_corners` argument of `F.interpolate`. Default: False.
- **loss_keypoint** (*dict*) – Config for loss. Default: None.

`forward(x)`

Forward function.

get_loss(*outputs, targets, masks*)
Calculate bottom-up masked mse loss.

Note:

- **batch_size**: N
 - **num_channels**: C
 - **heatmaps height**: H
 - **heatmaps weight**: W
-

Parameters

- **outputs** (*List(torch.Tensor[N, C, H, W])*) – Multi-scale outputs.
- **targets** (*List(torch.Tensor[N, C, H, W])*) – Multi-scale targets.
- **masks** (*List(torch.Tensor[N, H, W])*) – Masks of multi-scale targets.

init_weights()
Initialize model weights.

class mmpose.models.heads.**DeepposeRegressionHead**(*in_channels, num_joints, loss_keypoint=None, out_sigma=False, train_cfg=None, test_cfg=None*)

Deeppose regression head with fully connected layers.

“DeepPose: Human Pose Estimation via Deep Neural Networks”.

Parameters

- **in_channels** (*int*) – Number of input channels
- **num_joints** (*int*) – Number of joints
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.
- **out_sigma** (*bool*) – Predict the sigma (the viriance of the joint location) together with the joint location. Default: False

decode(*img metas, output, **kwargs*)
Decode the keypoints from output regression.

Parameters

- **img_metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - “image_file”: path to the image file
 - “center”: center of the bbox
 - “scale”: scale of the bbox
 - “rotation”: rotation of the bbox
 - “bbox_score”: score of bbox
- **output** (*np.ndarray[N, K, >=2]*) – predicted regression vector.
- **kwargs** – dict contains ‘img_size’. **img_size** (tuple(img_width, img_height)): input image size.

forward(*x*)

Forward function.

get_accuracy(*output*, *target*, *target_weight*)

Calculate accuracy for top-down keypoint loss.

Note:

- **batch_size**: N
 - **num_keypoints**: K
-

Parameters

- **output** (*torch.Tensor*[N, K, 2 or 4]) – Output keypoints.
- **target** (*torch.Tensor*[N, K, 2]) – Target keypoints.
- **target_weight** (*torch.Tensor*[N, K, 2]) – Weights across different joint types.

get_loss(*output*, *target*, *target_weight*)

Calculate top-down keypoint loss.

Note:

- **batch_size**: N
 - **num_keypoints**: K
-

Parameters

- **output** (*torch.Tensor*[N, K, 2 or 4]) – Output keypoints.
- **target** (*torch.Tensor*[N, K, 2]) – Target keypoints.
- **target_weight** (*torch.Tensor*[N, K, 2]) – Weights across different joint types.

inference_model(*x*, *flip_pairs*=None)

Inference function.

Returns Output regression.**Return type** output_regression (*np.ndarray*)**Parameters**

- **x** (*torch.Tensor*[N, K, 2]) – Input features.
- **flip_pairs** (None | *list[tuple()]*) – Pairs of keypoints which are mirrored.

class mmpose.models.heads.HMRMeshHead(*in_channels*, *simpl_mean_params*=None, *n_iter*=3)

SMPL parameters regressor head of simple baseline. “End-to-end Recovery of Human Shape and Pose”, CVPR’2018.

Parameters

- **in_channels** (*int*) – Number of input channels
- **simpl_mean_params** (*str*) – The file name of the mean SMPL parameters
- **n_iter** (*int*) – The iterations of estimating delta parameters

forward(*x*)

Forward function.

x is the image feature map and is expected to be in shape (batch size x channel number x height x width)

init_weights()

Initialize model weights.

```
class mmpose.models.heads.Interhand3DHead(keypoint_head_cfg, root_head_cfg, hand_type_head_cfg,
                                           loss_keypoint=None, loss_root_depth=None,
                                           loss_hand_type=None, train_cfg=None, test_cfg=None)
```

Interhand 3D head of paper ref: Gyeongsik Moon. “InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image”.

Parameters

- **keypoint_head_cfg** (*dict*) – Configs of Heatmap3DHead for hand keypoint estimation.
- **root_head_cfg** (*dict*) – Configs of Heatmap1DHead for relative hand root depth estimation.
- **hand_type_head_cfg** (*dict*) – Configs of MultilabelClassificationHead for hand type classification.
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.
- **loss_root_depth** (*dict*) – Config for relative root depth loss. Default: None.
- **loss_hand_type** (*dict*) – Config for hand type classification loss. Default: None.

decode(*img metas*, *output*, ***kwargs*)

Decode hand keypoint, relative root depth and hand type.

Parameters

- **img_metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - “image_file”: path to the image file
 - “center”: center of the bbox
 - “scale”: scale of the bbox
 - “rotation”: rotation of the bbox
 - “bbox_score”: score of bbox
 - “heatmap3d_depth_bound”: **depth bound of hand keypoint** 3D heatmap
 - “root_depth_bound”: **depth bound of relative root depth** 1D heatmap
- **output** (*list[np.ndarray]*) – model predicted 3D heatmaps, relative root depth and hand type.

forward(*x*)

Forward function.

get_accuracy(*output*, *target*, *target_weight*)

Calculate accuracy for hand type.

Parameters

- **output** (*list[Tensor]*) – a list of outputs from multiple heads.
- **target** (*list[Tensor]*) – a list of targets for multiple heads.

- **target_weight** (*list[[Tensor](#)]*) – a list of targets weight for multiple heads.

get_loss(*output, target, target_weight*)

Calculate loss for hand keypoint heatmaps, relative root depth and hand type.

Parameters

- **output** (*list[[Tensor](#)]*) – a list of outputs from multiple heads.
- **target** (*list[[Tensor](#)]*) – a list of targets for multiple heads.
- **target_weight** (*list[[Tensor](#)]*) – a list of targets weight for multiple heads.

inference_model(*x, flip_pairs=None*)

Inference function.

Returns list of output hand keypoint heatmaps, relative root depth and hand type.

Return type output (*list[[np.ndarray](#)]*)

Parameters

- **x** (*[torch.Tensor](#)[*N, K, H, W*]*) – Input features.
- **flip_pairs** (*None | list[tuple()]*) – Pairs of keypoints which are mirrored.

```
class mmpose.models.heads.MultiModalSSAHead(num_classes, modality, in_channels=1024,
                                             avg_pool_kernel=(1, 7, 7), dropout_prob=0.0,
                                             train_cfg=None, test_cfg=None, **kwargs)
```

Spatial-temporal Semantic Alignment Head proposed in “Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training”,

Please refer to the [paper](#) for details.

Parameters

- **num_classes** (*int*) – number of classes.
- **modality** (*list[str]*) – modalities of input videos for backbone.
- **in_channels** (*int*) – number of channels of feature maps. Default: 1024
- **avg_pool_kernel** (*tuple[int]*) – kernel size of pooling layer. Default: (1, 7, 7)
- **dropout_prob** (*float*) – probability to use dropout on input feature map. Default: 0
- **train_cfg** (*dict*) – training config.
- **test_cfg** (*dict*) – testing config.

forward(*x, img metas*)

Forward function.

get_accuracy(*logits, label, img metas*)

Compute the accuracy of predicted gesture.

Note:

- batch_size: N
 - number of classes: nC
 - logit length: L
-

Parameters

- **logits** (*list[NxnCxL]*) – predicted logits for each modality.
- **label** (*list(dict)*) – Category label.
- **img metas** (*list(dict)*) – Information about data. By default this includes: - “fps”: video frame rate - “modality”: modality of input videos

Returns computed accuracy for each modality.

Return type dict[str, torch.tensor]

get_loss(*logits, label, fmaps=None*)

Compute the Cross Entropy loss and SSA loss.

Note:

- batch_size: N
 - number of classes: nC
 - feature map channel: C
 - feature map height: H
 - feature map width: W
 - feature map length: L
 - logit length: Lg
-

Parameters

- **logits** (*list[NxnCxLg]*) – predicted logits for each modality.
- **label** (*list(dict)*) – Category label.
- **fmaps** (*list[torch.Tensor[NxCxLxHxW]]*) – feature maps for each modality.

Returns computed losses.

Return type dict[str, torch.tensor]

init_weights()

Initialize model weights.

set_train_epoch(*epoch: int*)

set the epoch to control the activation of SSA loss.

class mmpose.models.heads.**TemporalRegressionHead**(*in_channels, num_joints, max_norm=None, loss_keypoint=None, is_trajectory=False, train_cfg=None, test_cfg=None*)

Regression head of VideoPose3D.

“3D human pose estimation in video with temporal convolutions and semi-supervised training”, CVPR’2019.

Parameters

- **in_channels** (*int*) – Number of input channels
- **num_joints** (*int*) – Number of joints
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.

- **max_norm** (*float/None*) – if not None, the weight of convolution layers will be clipped to have a maximum norm of max_norm.
- **is_trajectory** (*bool*) – If the model only predicts root joint position, then this arg should be set to True. In this case, traj_loss will be calculated. Otherwise, it should be set to False. Default: False.

decode(*metas, output*)

Decode the keypoints from output regression.

Parameters

- **metas** (*list(dict)*) – Information about data augmentation. By default this includes:
 - "target_image_path": path to the image file
- **output** (*np.ndarray[N, K, 3]*) – predicted regression vector.
- **metas** – Information about data augmentation including:
 - target_image_path (str): Optional, path to the image file
 - **target_mean (float): Optional, normalization parameter of** the target pose.
 - **target_std (float): Optional, normalization parameter of the** target pose.
 - **root_position (np.ndarray[3,1]): Optional, global** position of the root joint.
 - **root_index (torch.ndarray[1,]): Optional, original index of** the root joint before root-centering.

forward(*x*)

Forward function.

get_accuracy(*output, target, target_weight, metas*)

Calculate accuracy for keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
-

Parameters

- **output** (*torch.Tensor[N, K, 3]*) – Output keypoints.
- **target** (*torch.Tensor[N, K, 3]*) – Target keypoints.
- **target_weight** (*torch.Tensor[N, K, 3]*) – Weights across different joint types.
- **metas** (*list(dict)*) – Information about data augmentation including:
 - target_image_path (str): Optional, path to the image file
 - **target_mean (float): Optional, normalization parameter of** the target pose.
 - **target_std (float): Optional, normalization parameter of the** target pose.
 - **root_position (np.ndarray[3,1]): Optional, global** position of the root joint.
 - **root_index (torch.ndarray[1,]): Optional, original index of** the root joint before root-centering.

get_loss(*output, target, target_weight*)
Calculate keypoint loss.

Note:

- **batch_size**: N
 - **num_keypoints**: K
-

Parameters

- **output** (*torch.Tensor[N, K, 3]*) – Output keypoints.
- **target** (*torch.Tensor[N, K, 3]*) – Target keypoints.
- **target_weight** (*torch.Tensor[N, K, 3]*) – Weights across different joint types. If `self.is_trajectory` is True and `target_weight` is None, `target_weight` will be set inversely proportional to joint depth.

inference_model(*x, flip_pairs=None*)
Inference function.

Returns Output regression.

Return type `output_regression` (`np.ndarray`)

Parameters

- **x** (*torch.Tensor[N, K, 2]*) – Input features.
- **flip_pairs** (*None | list[tuple()]*) – Pairs of keypoints which are mirrored.

init_weights()
Initialize the weights.

class `mmpose.models.heads.TopdownHeatmapBaseHead`
Base class for top-down heatmap heads.

All top-down heatmap heads should subclass it. All subclass should overwrite:

Methods:`get_loss`, supporting to calculate loss. Methods:`get_accuracy`, supporting to calculate accuracy. Methods:`forward`, supporting to forward model. Methods:`inference_model`, supporting to inference model.

decode(*img metas, output, **kwargs*)
Decode keypoints from heatmaps.

Parameters

- **img_metas** (*list(dict)*) – Information about data augmentation By default this includes:
 - `"image_file"`: path to the image file
 - `"center"`: center of the bbox
 - `"scale"`: scale of the bbox
 - `"rotation"`: rotation of the bbox
 - `"bbox_score"`: score of bbox
- **output** (*np.ndarray[N, K, H, W]*) – model predicted heatmaps.

abstract forward(**kwargs)

Forward function.

abstract get_accuracy(**kwargs)

Gets the accuracy.

abstract get_loss(**kwargs)

Gets the loss.

abstract inference_model(**kwargs)

Inference function.

```
class mmpose.models.heads.TopdownHeatmapMSMUHead(out_shape, unit_channels=256, out_channels=17,
                                                  num_stages=4, num_units=4, use_prm=False,
                                                  norm_cfg={'type': 'BN'}, loss_keypoint=None,
                                                  train_cfg=None, test_cfg=None)
```

Heads for multi-stage multi-unit heads used in Multi-Stage Pose estimation Network (MSPN), and Residual Steps Networks (RSN).

Parameters

- **unit_channels** (*int*) – Number of input channels.
- **out_channels** (*int*) – Number of output channels.
- **out_shape** (*tuple*) – Shape of the output heatmap.
- **num_stages** (*int*) – Number of stages.
- **num_units** (*int*) – Number of units in each stage.
- **use_prm** (*bool*) – Whether to use pose refine machine (PRM). Default: False.
- **norm_cfg** (*dict*) – dictionary to construct and config norm layer. Default: dict(type='BN')
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.

forward(*x*)

Forward function.

Returns

a list of heatmaps from multiple stages and units.

Return type out (list[*Tensor*])

get_accuracy(*output*, *target*, *target_weight*)

Calculate accuracy for top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor*[*N*, *K*, *H*, *W*]) – Output heatmaps.

- **target** (*torch.Tensor*[N, K, H, W]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[$N, K, 1$]) – Weights across different joint types.

get_loss(*output, target, target_weight*)
Calculate top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - num_outputs: O
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor*[N, O, K, H, W]) – Output heatmaps.
- **target** (*torch.Tensor*[N, O, K, H, W]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[$N, O, K, 1$]) – Weights across different joint types.

inference_model(*x, flip_pairs=None*)
Inference function.

Returns Output heatmaps.

Return type output_heatmap (np.ndarray)

Parameters

- **x** (*list*[*torch.Tensor*[N, K, H, W]]) – Input features.
- **flip_pairs** (*None* | *list*[*tuple*]) – Pairs of keypoints which are mirrored.

init_weights()
Initialize model weights.

```
class mmpose.models.heads.TopdownHeatmapMultiStageHead(in_channels=512, out_channels=17,  
                                                         num_stages=1, num_deconv_layers=3,  
                                                         num_deconv_filters=(256, 256, 256),  
                                                         num_deconv_kernels=(4, 4, 4), extra=None,  
                                                         loss_keypoint=None, train_cfg=None,  
                                                         test_cfg=None)
```

Top-down heatmap multi-stage head.

TopdownHeatmapMultiStageHead is consisted of multiple branches, each of which has num_deconv_layers(>=0) number of deconv layers and a simple conv2d layer.

Parameters

- **in_channels** (*int*) – Number of input channels.
- **out_channels** (*int*) – Number of output channels.
- **num_stages** (*int*) – Number of stages.

- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should \geq 0. Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers $>$ 0, the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.

forward(*x*)

Forward function.

Returns a list of heatmaps from multiple stages.

Return type out (list[*Tensor*])

get_accuracy(*output, target, target_weight*)

Calculate accuracy for top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor*[*N, K, H, W*]) – Output heatmaps.
- **target** (*torch.Tensor*[*N, K, H, W*]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[*N, K, 1*]) – Weights across different joint types.

get_loss(*output, target, target_weight*)

Calculate top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - num_outputs: O
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor*[*N, K, H, W*]) – Output heatmaps.
- **target** (*torch.Tensor*[*N, K, H, W*]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[*N, K, 1*]) – Weights across different joint types.

inference_model(*x*, *flip_pairs=None*)

Inference function.

Returns Output heatmaps.

Return type output_heatmap (np.ndarray)

Parameters

- **x** (*List[torch.Tensor[NxKxHxW]]*) – Input features.
- **flip_pairs** (*None* | *list[tuple()]*) – Pairs of keypoints which are mirrored.

init_weights()

Initialize model weights.

```
class mmpose.models.heads.TopdownHeatmapSimpleHead(in_channels, out_channels,
                                                    num_deconv_layers=3, num_deconv_filters=(256,
                                                    256, 256), num_deconv_kernels=(4, 4, 4),
                                                    extra=None, in_index=0, input_transform=None,
                                                    align_corners=False, loss_keypoint=None,
                                                    train_cfg=None, test_cfg=None)
```

Top-down heatmap simple head. paper ref: Bin Xiao et al. Simple Baselines for Human Pose Estimation and Tracking.

TopdownHeatmapSimpleHead is consisted of (≥ 0) number of deconv layers and a simple conv2d layer.

Parameters

- **in_channels** (*int*) – Number of input channels
- **out_channels** (*int*) – Number of output channels
- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should ≥ 0 . Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers > 0 , the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **in_index** (*int/Sequence[int]*) – Input feature index. Default: 0
- **input_transform** (*str/None*) – Transformation type of input features. Options: 'resize_concat', 'multiple_select', None. Default: None.
 - **'resize_concat'**: Multiple feature maps will be resized to the same size as the first one and then concat together. Usually used in FCN head of HRNet.
 - **'multiple_select'**: Multiple feature maps will be bundle into a list and passed into decode head.
 - None: Only one select feature map is allowed.
- **align_corners** (*bool*) – align_corners argument of F.interpolate. Default: False.
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.

forward(*x*)

Forward function.

get_accuracy(*output*, *target*, *target_weight*)

Calculate accuracy for top-down keypoint loss.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `heatmaps height`: H
 - `heatmaps weight`: W
-

Parameters

- **output** (*torch.Tensor*[N,K,H,W]) – Output heatmaps.
- **target** (*torch.Tensor*[N,K,H,W]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[N,K,1]) – Weights across different joint types.

get_loss(*output, target, target_weight*)
Calculate top-down keypoint loss.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `heatmaps height`: H
 - `heatmaps weight`: W
-

Parameters

- **output** (*torch.Tensor*[N,K,H,W]) – Output heatmaps.
- **target** (*torch.Tensor*[N,K,H,W]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[N,K,1]) – Weights across different joint types.

inference_model(*x, flip_pairs=None*)
Inference function.

Returns Output heatmaps.

Return type output_heatmap (np.ndarray)

Parameters

- **x** (*torch.Tensor*[N,K,H,W]) – Input features.
- **flip_pairs** (*None* | *list[tuple]*) – Pairs of keypoints which are mirrored.

init_weights()
Initialize model weights.

```
class mmpose.models.heads.ViPNASHeatmapSimpleHead(in_channels, out_channels, num_deconv_layers=3,
                                                    num_deconv_filters=(144, 144, 144),
                                                    num_deconv_kernels=(4, 4, 4),
                                                    num_deconv_groups=(16, 16, 16), extra=None,
                                                    in_index=0, input_transform=None,
                                                    align_corners=False, loss_keypoint=None,
                                                    train_cfg=None, test_cfg=None)
```

ViPNAS heatmap simple head.

ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search. More details can be found in the [paper](#).

TopdownHeatmapSimpleHead is consisted of (≥ 0) number of deconv layers and a simple conv2d layer.

Parameters

- **in_channels** (*int*) – Number of input channels
- **out_channels** (*int*) – Number of output channels
- **num_deconv_layers** (*int*) – Number of deconv layers. num_deconv_layers should ≥ 0 . Note that 0 means no deconv layers.
- **num_deconv_filters** (*list/tuple*) – Number of filters. If num_deconv_layers > 0 , the length of
- **num_deconv_kernels** (*list/tuple*) – Kernel sizes.
- **num_deconv_groups** (*list/tuple*) – Group number.
- **in_index** (*int/Sequence[int]*) – Input feature index. Default: -1
- **input_transform** (*str/None*) – Transformation type of input features. Options: 'resize_concat', 'multiple_select', None. Default: None.
 - **'resize_concat'**: Multiple feature maps will be resize to the same size as first one and then concat together. Usually used in FCN head of HRNet.
 - **'multiple_select'**: Multiple feature maps will be bundle into a list and passed into decode head.
 - None: Only one select feature map is allowed.
- **align_corners** (*bool*) – align_corners argument of F.interpolate. Default: False.
- **loss_keypoint** (*dict*) – Config for keypoint loss. Default: None.

forward(*x*)

Forward function.

get_accuracy(*output, target, target_weight*)

Calculate accuracy for top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor[N, K, H, W]*) – Output heatmaps.
- **target** (*torch.Tensor[N, K, H, W]*) – Target heatmaps.
- **target_weight** (*torch.Tensor[N, K, 1]*) – Weights across different joint types.

get_loss(*output, target, target_weight*)
Calculate top-down keypoint loss.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmaps height: H
 - heatmaps weight: W
-

Parameters

- **output** (*torch.Tensor[N, K, H, W]*) – Output heatmaps.
- **target** (*torch.Tensor[N, K, H, W]*) – Target heatmaps.
- **target_weight** (*torch.Tensor[N, K, 1]*) – Weights across different joint types.

inference_model(*x, flip_pairs=None*)
Inference function.

Returns Output heatmaps.

Return type output_heatmap (np.ndarray)

Parameters

- **x** (*torch.Tensor[N, K, H, W]*) – Input features.
- **flip_pairs** (*None | list[tuple]*) – Pairs of keypoints which are mirrored.

init_weights()
Initialize model weights.

49.5 losses

class mmpose.models.losses.**AELoss**(*loss_type*)
Associative Embedding loss.

Associative Embedding: End-to-End Learning for Joint Detection and Grouping.

forward(*tags, joints*)
Accumulate the tag loss for each image in the batch.

Note:

- batch_size: N
 - heatmaps weight: W
 - heatmaps height: H
 - max_num_people: M
 - num_keypoints: K
-

Parameters

- **tags** (*torch.Tensor*[*N*,*KxHxW*,1]) – tag channels of output.
- **joints** (*torch.Tensor*[*N*,*M*,*K*,2]) – joints information.

singleTagLoss(*pred_tag*, *joints*)

Associative embedding loss for one image.

Note:

- heatmaps weight: W
- heatmaps height: H
- max_num_people: M
- num_keypoints: K

Parameters

- **pred_tag** (*torch.Tensor*[*KxHxW*,1]) – tag of output for one image.
- **joints** (*torch.Tensor*[*M*,*K*,2]) – joints information for one image.

class mmpose.models.losses.**AdaptiveWingLoss**(*alpha=2.1*, *omega=14*, *epsilon=1*, *theta=0.5*,
use_target_weight=False, *loss_weight=1.0*)

Adaptive wing loss. paper ref: ‘Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression’ Wang et al. ICCV’2019.

Parameters

- **alpha** (*float*), **omega** (*float*), **epsilon** (*float*), **theta** (*float*) – are hyper-parameters.
- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

criterion(*pred*, *target*)

Criterion of wingloss.

Note: batch_size: N num_keypoints: K

Parameters

- **pred** (*torch.Tensor*[*NxKxHxW*]) – Predicted heatmaps.
- **target** (*torch.Tensor*[*NxKxHxW*]) – Target heatmaps.

forward(*output*, *target*, *target_weight*)

Forward function.

Note: batch_size: N num_keypoints: K

Parameters

- **output** (*torch.Tensor*[*NxKxHxW*]) – Output heatmaps.
- **target** (*torch.Tensor*[*NxKxHxW*]) – Target heatmaps.
- **target_weight** (*torch.Tensor*[*NxKx1*]) – Weights across different joint types.

class `mmpose.models.losses.BCELoss`(*use_target_weight=False, loss_weight=1.0*)

Binary Cross Entropy loss.

forward(*output, target, target_weight=None*)

Forward function.

Note:

- **batch_size**: N
 - **num_labels**: K
-

Parameters

- **output** (*torch.Tensor*[*N, K*]) – Output classification.
- **target** (*torch.Tensor*[*N, K*]) – Target classification.
- **target_weight** (*torch.Tensor*[*N, K*] or *torch.Tensor*[*N*]) – Weights across different labels.

class `mmpose.models.losses.BoneLoss`(*joint_parents, use_target_weight=False, loss_weight=1.0*)

Bone length loss.

Parameters

- **joint_parents** (*list*) – Indices of each joint's parent joint.
- **use_target_weight** (*bool*) – Option to use weighted bone loss. Different bone types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output, target, target_weight=None*)

Forward function.

Note:

- **batch_size**: N
 - **num_keypoints**: K
 - **dimension of keypoints**: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor*[*N, K, D*]) – Output regression.
- **target** (*torch.Tensor*[*N, K, D*]) – Target regression.
- **target_weight** (*torch.Tensor*[*N, K-1*]) – Weights across different bone types.

class `mmpose.models.losses.FocalHeatmapLoss`(*alpha=2, beta=4*)

forward(*pred, gt, mask=None*)
Modified focal loss.

Exactly the same as CornerNet. Runs faster and costs a little bit more memory :param pred: :type pred: batch x c x h x w :param gt_regr: :type gt_regr: batch x c x h x w

class mmpose.models.losses.**GANLoss**(*gan_type, real_label_val=1.0, fake_label_val=0.0, loss_weight=1.0*)
Define GAN loss.

Parameters

- **gan_type** (*str*) – Support ‘vanilla’, ‘lsgan’, ‘wgan’, ‘hinge’.
- **real_label_val** (*float*) – The value for real label. Default: 1.0.
- **fake_label_val** (*float*) – The value for fake label. Default: 0.0.
- **loss_weight** (*float*) – Loss weight. Default: 1.0. Note that loss_weight is only for generators; and it is always 1.0 for discriminators.

forward(*input, target_is_real, is_disc=False*)

Parameters

- **input** (*Tensor*) – The input for the loss module, i.e., the network prediction.
- **target_is_real** (*bool*) – Whether the target is real or fake.
- **is_disc** (*bool*) – Whether the loss for discriminators or not. Default: False.

Returns GAN loss value.

Return type Tensor

get_target_label(*input, target_is_real*)
Get target label.

Parameters

- **input** (*Tensor*) – Input tensor.
- **target_is_real** (*bool*) – Whether the target is real or fake.

Returns Target tensor. Return bool for wgan, otherwise, return Tensor.

Return type (bool | Tensor)

class mmpose.models.losses.**HeatmapLoss**(*supervise_empty=True*)
Accumulate the heatmap loss for each image in the batch.

Parameters **supervise_empty** (*bool*) – Whether to supervise empty channels.

forward(*pred, gt, mask*)
Forward function.

Note:

- batch_size: N
- heatmaps weight: W
- heatmaps height: H
- max_num_people: M
- num_keypoints: K

Parameters

- **pred** (*torch.Tensor*[*N*,*K*,*H*,*W*]) – heatmap of output.
- **gt** (*torch.Tensor*[*N*,*K*,*H*,*W*]) – target heatmap.
- **mask** (*torch.Tensor*[*N*,*H*,*W*]) – mask of target.

class `mmpose.models.losses.JointsMSELoss`(*use_target_weight=False*, *loss_weight=1.0*)
MSE loss for heatmaps.

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output*, *target*, *target_weight*)
Forward function.

class `mmpose.models.losses.JointsOHKMMSELoss`(*use_target_weight=False*, *topk=8*, *loss_weight=1.0*)
MSE loss with online hard keypoint mining.

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **topk** (*int*) – Only top k joint losses are kept.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output*, *target*, *target_weight*)
Forward function.

class `mmpose.models.losses.L1Loss`(*use_target_weight=False*, *loss_weight=1.0*)
L1Loss loss .

forward(*output*, *target*, *target_weight=None*)
Forward function.

Note:

- batch_size: N
 - num_keypoints: K
-

Parameters

- **output** (*torch.Tensor*[*N*, *K*, 2]) – Output regression.
- **target** (*torch.Tensor*[*N*, *K*, 2]) – Target regression.
- **target_weight** (*torch.Tensor*[*N*, *K*, 2]) – Weights across different joint types.

class `mmpose.models.losses.MPJPELoss`(*use_target_weight=False*, *loss_weight=1.0*)
MPJPE (Mean Per Joint Position Error) loss.

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output, target, target_weight=None*)
Forward function.

Note:

- batch_size: N
 - num_keypoints: K
 - dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor[N, K, D]*) – Output regression.
- **target** (*torch.Tensor[N, K, D]*) – Target regression.
- **target_weight** (*torch.Tensor[N, K, D]*) – Weights across different joint types.

class `mmpose.models.losses.MSELoss`(*use_target_weight=False, loss_weight=1.0*)
MSE loss for coordinate regression.

forward(*output, target, target_weight=None*)
Forward function.

Note:

- batch_size: N
 - num_keypoints: K
-

Parameters

- **output** (*torch.Tensor[N, K, 2]*) – Output regression.
- **target** (*torch.Tensor[N, K, 2]*) – Target regression.
- **target_weight** (*torch.Tensor[N, K, 2]*) – Weights across different joint types.

class `mmpose.models.losses.MeshLoss`(*joints_2d_loss_weight, joints_3d_loss_weight, vertex_loss_weight, smpl_pose_loss_weight, smpl_beta_loss_weight, img_res, focal_length=5000*)

Mix loss for 3D human mesh. It is composed of loss on 2D joints, 3D joints, mesh vertices and smpl parameters (if any).

Parameters

- **joints_2d_loss_weight** (*float*) – Weight for loss on 2D joints.
- **joints_3d_loss_weight** (*float*) – Weight for loss on 3D joints.
- **vertex_loss_weight** (*float*) – Weight for loss on 3D vertices.
- **smpl_pose_loss_weight** (*float*) – Weight for loss on SMPL pose parameters.
- **smpl_beta_loss_weight** (*float*) – Weight for loss on SMPL shape parameters.

- **img_res** (*int*) – Input image resolution.
- **focal_length** (*float*) – Focal length of camera model. Default=5000.

forward(*output, target*)

Forward function.

Parameters

- **output** (*dict*) – dict of network predicted results. Keys: ‘vertices’, ‘joints_3d’, ‘camera’, ‘pose’(optional), ‘beta’(optional)
- **target** (*dict*) – dict of ground-truth labels. Keys: ‘vertices’, ‘joints_3d’, ‘joints_3d_visible’, ‘joints_2d’, ‘joints_2d_visible’, ‘pose’, ‘beta’, ‘has_smpl’

Returns dict of losses.

Return type dict

joints_2d_loss(*pred_joints_2d, gt_joints_2d, joints_2d_visible*)

Compute 2D reprojection loss on the joints.

The loss is weighted by joints_2d_visible.

joints_3d_loss(*pred_joints_3d, gt_joints_3d, joints_3d_visible*)

Compute 3D joints loss for the examples that 3D joint annotations are available.

The loss is weighted by joints_3d_visible.

project_points(*points_3d, camera*)

Perform orthographic projection of 3D points using the camera parameters, return projected 2D points in image plane.

Note:

- batch size: B
 - point number: N
-

Parameters

- **points_3d** (*Tensor([B, N, 3])*) – 3D points.
- **camera** (*Tensor([B, 3])*) – camera parameters with the 3 channel as (scale, translation_x, translation_y)

Returns projected 2D points in image space.

Return type Tensor([B, N, 2])

smpl_losses(*pred_rotmat, pred_betas, gt_pose, gt_betas, has_smpl*)

Compute SMPL parameters loss for the examples that SMPL parameter annotations are available.

The loss is weighted by has_smpl.

vertex_loss(*pred_vertices, gt_vertices, has_smpl*)

Compute 3D vertex loss for the examples that 3D human mesh annotations are available.

The loss is weighted by the has_smpl.

```
class mmpose.models.losses.MultiLossFactory(num_joints, num_stages, ae_loss_type, with_ae_loss,
                                             push_loss_factor, pull_loss_factor, with_heatmaps_loss,
                                             heatmaps_loss_factor, supervise_empty=True)
```

Loss for bottom-up models.

Parameters

- **num_joints** (*int*) – Number of keypoints.
- **num_stages** (*int*) – Number of stages.
- **ae_loss_type** (*str*) – Type of ae loss.
- **with_ae_loss** (*list[bool]*) – Use ae loss or not in multi-heatmap.
- **push_loss_factor** (*list[float]*) – Parameter of push loss in multi-heatmap.
- **pull_loss_factor** (*list[float]*) – Parameter of pull loss in multi-heatmap.
- **with_heatmap_loss** (*list[bool]*) – Use heatmap loss or not in multi-heatmap.
- **heatmaps_loss_factor** (*list[float]*) – Parameter of heatmap loss in multi-heatmap.
- **supervise_empty** (*bool*) – Whether to supervise empty channels.

forward(*outputs, heatmaps, masks, joints*)

Forward function to calculate losses.

Note:

- batch_size: N
 - heatmaps weight: W
 - heatmaps height: H
 - max_num_people: M
 - num_keypoints: K
 - output_channel: C C=2K if use ae loss else K
-

Parameters

- **outputs** (*list(torch.Tensor[N, C, H, W])*) – outputs of stages.
- **heatmaps** (*list(torch.Tensor[N, K, H, W])*) – target of heatmaps.
- **masks** (*list(torch.Tensor[N, H, W])*) – masks of heatmaps.
- **joints** (*list(torch.Tensor[N, M, K, 2])*) – joints of ae loss.

```
class mmpose.models.losses.RLELoss(use_target_weight=False, size_average=True, residual=True,
                                    q_dis='laplace')
```

RLE Loss.

[Human Pose Regression With Residual Log-Likelihood Estimation arXiv.](#)

Code is modified from [the official implementation](#).

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.

- **size_average** (*bool*) – Option to average the loss by the batch_size.
- **residual** (*bool*) – Option to add L1 loss and let the flow learn the residual error distribution.
- **q_dis** (*string*) – Option for the identity Q(error) distribution, Options: “laplace” or “gaussian”

forward(*output, target, target_weight=None*)

Forward function.

Note:

- batch_size: N
 - num_keypoints: K
 - dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor[N, K, D*2]*) – Output regression, including coords and sigmas.
- **target** (*torch.Tensor[N, K, D]*) – Target regression.
- **target_weight** (*torch.Tensor[N, K, D]*) – Weights across different joint types.

class mmpose.models.losses.SemiSupervisionLoss(*joint_parents, projection_loss_weight=1.0, bone_loss_weight=1.0, warmup_iterations=0*)

Semi-supervision loss for unlabeled data. It is composed of projection loss and bone loss.

Paper ref: *3D human pose estimation in video with temporal convolutions and semi-supervised training* Dario Pavlo et al. CVPR’2019.

Parameters

- **joint_parents** (*list*) – Indices of each joint’s parent joint.
- **projection_loss_weight** (*float*) – Weight for projection loss.
- **bone_loss_weight** (*float*) – Weight for bone loss.
- **warmup_iterations** (*int*) – Number of warmup iterations. In the first *warmup_iterations* iterations, the model is trained only on labeled data, and semi-supervision loss will be 0. This is a workaround since currently we cannot access epoch number in loss functions. Note that the iteration number in an epoch can be changed due to different GPU numbers in multi-GPU settings. So please set this parameter carefully.
warmup_iterations = dataset_size // samples_per_gpu // gpu_num * warmup_epochs

forward(*output, target*)

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note: Although the recipe for forward pass needs to be defined within this function, one should call the Module instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

static project_joints(*x*, *intrinsic*s)

Project 3D joint coordinates to 2D image plane using camera intrinsic parameters.

Parameters

- **x** (*torch.Tensor*[*N*, *K*, 3]) – 3D joint coordinates.
- **intrinsic**s (*torch.Tensor*[*N*, 4] | *torch.Tensor*[*N*, 9]) – Camera intrinsic: f (2), c (2), k (3), p (2).

class mmpose.models.losses.**SmoothL1Loss**(*use_target_weight=False*, *loss_weight=1.0*)

SmoothL1Loss loss.

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output*, *target*, *target_weight=None*)

Forward function.

Note:

- batch_size: N
 - num_keypoints: K
 - dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor*[*N*, *K*, *D*]) – Output regression.
- **target** (*torch.Tensor*[*N*, *K*, *D*]) – Target regression.
- **target_weight** (*torch.Tensor*[*N*, *K*, *D*]) – Weights across different joint types.

class mmpose.models.losses.**SoftWeightSmoothL1Loss**(*use_target_weight=False*, *supervise_empty=True*, *beta=1.0*, *loss_weight=1.0*)

Smooth L1 loss with soft weight for regression.

Parameters

- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **supervise_empty** (*bool*) – Whether to supervise the output with zero weight.
- **beta** (*float*) – Specifies the threshold at which to change between L1 and L2 loss.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

forward(*output*, *target*, *target_weight=None*)

Forward function.

Note:

- batch_size: N
 - num_keypoints: K
-

- dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor*[N, K, D]) – Output regression.
- **target** (*torch.Tensor*[N, K, D]) – Target regression.
- **target_weight** (*torch.Tensor*[N, K, D]) – Weights across different joint types.

static smooth_l1_loss(*input, target, reduction='none', beta=1.0*)

Re-implement torch.nn.functional.smooth_l1_loss with beta to support pytorch <= 1.6.

class mmpose.models.losses.**SoftWingLoss**(*omega1=2.0, omega2=20.0, epsilon=0.5,*
use_target_weight=False, loss_weight=1.0)

Soft Wing Loss ‘Structure-Coherent Deep Feature Learning for Robust Face Alignment’ Lin et al. TIP’2021.

loss =

1. $|x|$, if $|x| < \text{omega1}$
2. $\text{omega2} * \ln(1 + |x| / \text{epsilon}) + B$, if $|x| \geq \text{omega1}$

Parameters

- **omega1** (*float*) – The first threshold.
- **omega2** (*float*) – The second threshold.
- **epsilon** (*float*) – Also referred to as curvature.
- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

criterion(*pred, target*)

Criterion of wingloss.

Note: batch_size: N num_keypoints: K dimension of keypoints: D (D=2 or D=3)

Parameters

- **pred** (*torch.Tensor*[N, K, D]) – Output regression.
- **target** (*torch.Tensor*[N, K, D]) – Target regression.

forward(*output, target, target_weight=None*)

Forward function.

Note: batch_size: N num_keypoints: K dimension of keypoints: D (D=2 or D=3)

Parameters

- **output** (*torch.Tensor*[N, K, D]) – Output regression.
- **target** (*torch.Tensor*[N, K, D]) – Target regression.

- **target_weight** (*torch.Tensor*[*N*, *K*, *D*]) – Weights across different joint types.

class mmpose.models.losses.**WingLoss**(*omega=10.0*, *epsilon=2.0*, *use_target_weight=False*, *loss_weight=1.0*)

Wing Loss. paper ref: ‘Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural Networks’ Feng et al. CVPR’2018.

Parameters

- **omega** (*float*) – Also referred to as width.
- **epsilon** (*float*) – Also referred to as curvature.
- **use_target_weight** (*bool*) – Option to use weighted MSE loss. Different joint types may have different target weights.
- **loss_weight** (*float*) – Weight of the loss. Default: 1.0.

criterion(*pred*, *target*)

Criterion of wingloss.

Note:

- batch_size: N
 - num_keypoints: K
 - dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **pred** (*torch.Tensor*[*N*, *K*, *D*]) – Output regression.
- **target** (*torch.Tensor*[*N*, *K*, *D*]) – Target regression.

forward(*output*, *target*, *target_weight=None*)

Forward function.

Note:

- batch_size: N
 - num_keypoints: K
 - dimension of keypoints: D (D=2 or D=3)
-

Parameters

- **output** (*torch.Tensor*[*N*, *K*, *D*]) – Output regression.
- **target** (*torch.Tensor*[*N*, *K*, *D*]) – Target regression.
- **target_weight** (*torch.Tensor*[*N*, *K*, *D*]) – Weights across different joint types.

49.6 misc

MMPOSE.DATASETS

```
class mmpose.datasets.AnimalATRWDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,  
                                         test_mode=False)
```

ATRW dataset for animal pose estimation.

“ATRW: A Benchmark for Amur Tiger Re-identification in the Wild” ACM MM’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

ATRW keypoint indexes:

```
0: "left_ear",
1: "right_ear",
2: "nose",
3: "right_shoulder",
4: "right_front_paw",
5: "left_shoulder",
6: "left_front_paw",
7: "right_hip",
8: "right_knee",
9: "right_back_paw",
10: "left_hip",
11: "left_knee",
12: "left_back_paw",
13: "tail",
14: "center"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
evaluate(results, res_folder=None, metric='mAP', **kwargs)
```

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (np.ndarray[N,K,3]): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (np.ndarray[N,6]): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (list[str]): For example, ['data/coco/val2017/000000393226.jpg']
 - heatmap (np.ndarray[N, K, H, W]): model output heatmap
 - bbox_id (list(int)).
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str / list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**AnimalFlyDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

AnimalFlyDataset for animal pose estimation.

“Fast animal pose estimation using deep neural networks” Nature methods’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Vinegar Fly keypoint indexes:

```
0: "head",
1: "eyeL",
2: "eyeR",
3: "neck",
4: "thorax",
5: "abdomen",
6: "forelegR1",
7: "forelegR2",
8: "forelegR3",
9: "forelegR4",
10: "midlegR1",
11: "midlegR2",
12: "midlegR3",
13: "midlegR4",
```

(continues on next page)

(continued from previous page)

```

14: "hindlegR1",
15: "hindlegR2",
16: "hindlegR3",
17: "hindlegR4",
18: "forelegL1",
19: "forelegL2",
20: "forelegL3",
21: "forelegL4",
22: "midlegL1",
23: "midlegL2",
24: "midlegL3",
25: "midlegL4",
26: "hindlegL1",
27: "hindlegL2",
28: "hindlegL3",
29: "hindlegL4",
30: "wingL",
31: "wingR"

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate Fly keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['Test/source/0.jpg']

- `output_heatmap` (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (`str`) – Path of directory to save the results.
- **metric** (`str` | `list[str]`) – Metric to be performed. Options: ‘PCK’, ‘AUC’, ‘EPE’.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.AnimalHorse10Dataset`(`ann_file`, `img_prefix`, `data_cfg`, `pipeline`, `dataset_info=None`, `test_mode=False`)

AnimalHorse10Dataset for animal pose estimation.

“Pretraining boosts out-of-domain robustness for pose estimation” WACV’2021. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Horse-10 keypoint indexes:

```
0: 'Nose',
1: 'Eye',
2: 'Neckknee',
3: 'Nearfrontfetlock',
4: 'Nearfrontfoot',
5: 'Offknee',
6: 'Offfrontfetlock',
7: 'Offfrontfoot',
8: 'Shoulder',
9: 'Midshoulder',
10: 'Elbow',
11: 'Girth',
12: 'Wither',
13: 'Nearhindhock',
14: 'Nearhindfetlock',
15: 'Nearhindfoot',
16: 'Hip',
17: 'Stifle',
18: 'Offhindhock',
19: 'Offhindfetlock',
20: 'Offhindfoot',
21: 'Ischium'
```

Parameters

- **ann_file** (`str`) – Path to the annotation file.
- **img_prefix** (`str`) – Path to a directory where images are held. Default: None.
- **data_cfg** (`dict`) – config
- **pipeline** (`list[dict` | `callable]`) – A sequence of data transforms.
- **dataset_info** (`DatasetInfo`) – A class containing all dataset info.
- **test_mode** (`bool`) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate horse-10 keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (`np.ndarray[N,6]`): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['Test/source/0.jpg']
 - output_heatmap (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* / *list[str]*) – Metric to be performed. Options: 'PCK', 'NME'.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**AnimalLocustDataset**(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

AnimalLocustDataset for animal pose estimation.

“DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning” Elife’2019. More details can be found in the paper.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Desert Locust keypoint indexes:

```
0: "head",
1: "neck",
2: "thorax",
3: "abdomen1",
4: "abdomen2",
5: "anttipL",
6: "antbaseL",
7: "eyeL",
8: "forelegL1",
9: "forelegL2",
10: "forelegL3",
11: "forelegL4",
```

(continues on next page)

(continued from previous page)

```

12: "midlegL1",
13: "midlegL2",
14: "midlegL3",
15: "midlegL4",
16: "hindlegL1",
17: "hindlegL2",
18: "hindlegL3",
19: "hindlegL4",
20: "anttipR",
21: "antbaseR",
22: "eyeR",
23: "forelegR1",
24: "forelegR2",
25: "forelegR3",
26: "forelegR4",
27: "midlegR1",
28: "midlegR2",
29: "midlegR3",
30: "midlegR4",
31: "hindlegR1",
32: "hindlegR2",
33: "hindlegR3",
34: "hindlegR4"

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate Fly keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:

- `preds` (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
- `boxes` (`np.ndarray[N,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
- `image_paths` (`list[str]`): For example, [`'Test/source/0.jpg'`]
- `output_heatmap` (`np.ndarray[N, K, H, W]`): model outputs.
- **`res_folder`** (`str`, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **`metric`** (`str` | `list[str]`) – Metric to be performed. Options: `'PCK'`, `'AUC'`, `'EPE'`.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.AnimalMacaqueDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info*=`None`, *test_mode*=`False`)

MacaquePose dataset for animal pose estimation.

“MacaquePose: A novel ‘in the wild’ macaque monkey pose dataset for markerless motion capture” bioRxiv’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Macaque keypoint indexes:

```
0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'
```

Parameters

- **`ann_file`** (`str`) – Path to the annotation file.
- **`img_prefix`** (`str`) – Path to a directory where images are held. Default: `None`.
- **`data_cfg`** (`dict`) – config
- **`pipeline`** (`list[dict` | `callable]`) – A sequence of data transforms.
- **`dataset_info`** (`DatasetInfo`) – A class containing all dataset info.
- **`test_mode`** (`bool`) – Store True when building test or validation dataset. Default: `False`.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note: batch_size: N num_keypoints: K heatmap height: H heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (`np.ndarray[N,6]`): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['data/coco/val2017/000000393226.jpg']
 - heatmap (`np.ndarray[N, K, H, W]`): model output heatmap
 - bbox_id (*list(int)*).
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.AnimalPoseDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

Animal-Pose dataset for animal pose estimation.

“Cross-domain Adaptation For Animal Pose Estimation” ICCV’2019 More details can be found in the [paper](#) .

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Animal-Pose keypoint indexes:

```
0: 'L_Eye',
1: 'R_Eye',
2: 'L_EarBase',
3: 'R_EarBase',
4: 'Nose',
5: 'Throat',
6: 'TailBase',
7: 'Withers',
8: 'L_F_Elbow',
9: 'R_F_Elbow',
10: 'L_B_Elbow',
11: 'R_B_Elbow',
12: 'L_F_Knee',
13: 'R_F_Knee',
14: 'L_B_Knee',
15: 'R_B_Knee',
16: 'L_F_Paw',
```

(continues on next page)

(continued from previous page)

```

17: 'R_F_Paw',
18: 'L_B_Paw',
19: 'R_B_Paw'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['data/coco/val2017/000000393226.jpg']
 - **heatmap** (*np.ndarray[N, K, H, W]*): model output heatmap
 - **bbox_id** (*list(int)*).
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

```

class mmpose.datasets.AnimalZebraDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,
                                         test_mode=False)

```

AnimalZebraDataset for animal pose estimation.

“DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning” Elife’2019. More details can be found in the paper.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Desert Locust keypoint indexes:

```
0: "snout",
1: "head",
2: "neck",
3: "forelegL1",
4: "forelegR1",
5: "hindlegL1",
6: "hindlegR1",
7: "tailbase",
8: "tailtip"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate Fly keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['Test/source/0.jpg']
 - output_heatmap (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.

- **metric** (*str* | *list[str]*) – Metric to be performed. Options: ‘PCK’, ‘AUC’, ‘EPE’.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**Body3DH36MDataset**(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info*=None, *test_mode*=False)

Human3.6M dataset for 3D human pose estimation.

“Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”, TPAMI 2014. More details can be found in the [paper](#).

Human3.6M keypoint indexes:

```
0: 'root (pelvis)',
1: 'right_hip',
2: 'right_knee',
3: 'right_foot',
4: 'left_hip',
5: 'left_knee',
6: 'left_foot',
7: 'spine',
8: 'thorax',
9: 'neck_base',
10: 'head',
11: 'left_shoulder',
12: 'left_elbow',
13: 'left_wrist',
14: 'right_shoulder',
15: 'right_elbow',
16: 'right_wrist'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

build_sample_indices()

Split original videos into sequences and build frame indices.

This method overrides the default one in the base class.

evaluate(*results*, *res_folder*=None, *metric*='mpjpe', ***kwargs*)

Evaluate keypoint results.

get_camera_param(*imgname*)

Get camera parameters of a frame by its image name.

load_annotations()

Load data annotation.

load_config(data_cfg)

Initialize dataset attributes according to the config.

Override this method to set dataset specific attributes.

class mmpose.datasets.**Body3DMviewDirectCampusDataset**(ann_file, img_prefix, data_cfg, pipeline,
dataset_info=None, test_mode=False)

Campus dataset for direct multi-view human pose estimation.

*3D Pictorial Structures for Multiple Human Pose Estimation' CVPR'2014 More details can be found in the paper
<<http://campar.in.tum.de/pub/belagiannis2014cvpr/belagiannis2014cvpr.pdf>>*

The dataset loads both 2D and 3D annotations as well as camera parameters. It is worth mentioning that when training multi-view 3D pose models, due to the limited and incomplete annotations of this dataset, we may not use this dataset to train the model. Instead, we use the 2D pose estimator trained on COCO, and use independent 3D human poses from the CMU Panoptic dataset to train the 3D model. For testing, we first estimate 2D poses and generate 2D heatmaps for this dataset as the input to 3D model.

Campus keypoint indices:

```
'Right-Ankle': 0,  
'Right-Knee': 1,  
'Right-Hip': 2,  
'Left-Hip': 3,  
'Left-Knee': 4,  
'Left-Ankle': 5,  
'Right-Wrist': 6,  
'Right-Elbow': 7,  
'Right-Shoulder': 8,  
'Left-Shoulder': 9,  
'Left-Elbow': 10,  
'Left-Wrist': 11,  
'Bottom-Head': 12,  
'Top-Head': 13,
```

Parameters

- **ann_file** (str) – Path to the annotation file.
- **img_prefix** (str) – Path to a directory where images are held. Default: None.
- **data_cfg** (dict) – config
- **pipeline** (list[dict | callable]) – A sequence of data transforms.
- **dataset_info** (DatasetInfo) – A class containing all dataset info.
- **test_mode** (bool) – Store True when building test or validation dataset. Default: False.

static coco2campus3D(coco_pose)

transform coco order(our method output) 3d pose to campus dataset order with interpolation.

Parameters coco_pose – np.array with shape 17x3

Returns: 3D pose in campus order with shape 14x3

evaluate(results, res_folder=None, metric='pcp', recall_threshold=500, alpha_error=0.5, **kwargs)

Parameters

- **results** (*list[dict]*) – Testing results containing the following items: - pose_3d (np.ndarray): predicted 3D human pose - sample_id (np.ndarray): sample id of a frame.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'pcp'.
- **recall_threshold** – threshold for calculating recall.
- **alpha_error** – coefficient when calculating error for correct parts.
- ****kwargs** –

Returns:

static get_new_center(*center_list*)

Generate new center or select from the center list randomly.

The probability and the parameters related to coordinates can also be tuned, just make sure that the center is within the given 3D space.

isvalid(*new_center, bbox, bbox_list*)

Check if the new person bbox are valid, which need to satisfies:

1. the center is visible in at least 2 views, and
2. have a sufficiently small iou with all other person bboxes.

load_config(*data_cfg*)

Initialize dataset attributes according to the config.

Override this method to set dataset specific attributes.

class mmpose.datasets.**Body3DMviewDirectPanopticDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

Panoptic dataset for direct multi-view human pose estimation.

Panoptic Studio: A Massively Multiview System for Social Motion Capture' ICCV'2015 More details can be found in the `paper` .

The dataset loads both 2D and 3D annotations as well as camera parameters.

Panoptic keypoint indexes:

```
'neck': 0,
'nose': 1,
'mid-hip': 2,
'l-shoulder': 3,
'l-elbow': 4,
'l-wrist': 5,
'l-hip': 6,
'l-knee': 7,
'l-ankle': 8,
'r-shoulder': 9,
'r-elbow': 10,
'r-wrist': 11,
'r-hip': 12,
'r-knee': 13,
'r-ankle': 14,
'l-eye': 15,
```

(continues on next page)

(continued from previous page)

```
'l-ear': 16,
'r-eye': 17,
'r-ear': 18,
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mpjpe'*, ***kwargs*)

Parameters

- **results** (*list[dict]*) – Testing results containing the following items: - pose_3d (np.ndarray): predicted 3D human pose - sample_id (np.ndarray): sample id of a frame.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mpjpe'.
- ****kwargs** –

Returns:

load_config(*data_cfg*)

Initialize dataset attributes according to the config.

Override this method to set dataset specific attributes.

class mmpose.datasets.**Body3DmviewDirectShelfDataset**(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

Shelf dataset for direct multi-view human pose estimation.

3D Pictorial Structures for Multiple Human Pose Estimation' CVPR'2014 More details can be found in the paper <<http://campar.in.tum.de/pub/belagiannis2014cvpr/belagiannis2014cvpr.pdf>>

The dataset loads both 2D and 3D annotations as well as camera parameters. It is worth mentioning that when training multi-view 3D pose models, due to the limited and incomplete annotations of this dataset, we may not use this dataset to train the model. Instead, we use the 2D pose estimator trained on COCO, and use independent 3D human poses from the CMU Panoptic dataset to train the 3D model. For testing, we first estimate 2D poses and generate 2D heatmaps for this dataset as the input to 3D model.

Shelf keypoint indices:

```
'Right-Ankle': 0,
'Right-Knee': 1,
'Right-Hip': 2,
'Left-Hip': 3,
'Left-Knee': 4,
```

(continues on next page)

(continued from previous page)

```
'Left-Ankle': 5,
'Right-Wrist': 6,
'Right-Elbow': 7,
'Right-Shoulder': 8,
'Left-Shoulder': 9,
'Left-Elbow': 10,
'Left-Wrist': 11,
'Bottom-Head': 12,
'Top-Head': 13,
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

static coco2shelf3D(*coco_pose*, *alpha*=0.75)

transform coco order(our method output) 3d pose to shelf dataset order with interpolation.

Parameters *coco_pose* – np.array with shape 17x3

Returns: 3D pose in shelf order with shape 14x3

evaluate(*results*, *res_folder*=None, *metric*='pcp', *recall_threshold*=500, *alpha_error*=0.5, *alpha_head*=0.75, ***kwargs*)

Parameters

- **results** (*list[dict]*) – Testing results containing the following items: - *pose_3d* (np.ndarray): predicted 3D human pose - *sample_id* (np.ndarray): sample id of a frame.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'pcp'.
- **recall_threshold** – threshold for calculating recall.
- **alpha_error** – coefficient when calculating correct parts.
- **alpha_head** – coefficient for computing head keypoints position when converting coco poses to shelf poses
- ****kwargs** –

Returns:

static get_new_center(*center_list*)

Generate new center or select from the center list randomly.

The probability and the parameters related to coordinates can also be tuned, just make sure that the center is within the given 3D space.

static `isvalid(bbox, bbox_list)`

Check if the new person bbox are valid, which need to satisfies:

have a sufficiently small iou with all other person bboxes.

load_config(`data_cfg`)

Initialize dataset attributes according to the config.

Override this method to set dataset specific attributes.

class `mmpose.datasets.BottomUpAicDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False)`

Aic dataset for bottom-up pose estimation.

“AI Challenger : A Large-scale Dataset for Going Deeper in Image Understanding”, arXiv’2017. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

AIC keypoint indexes:

```
0: "right_shoulder",
1: "right_elbow",
2: "right_wrist",
3: "left_shoulder",
4: "left_elbow",
5: "left_wrist",
6: "right_hip",
7: "right_knee",
8: "right_ankle",
9: "left_hip",
10: "left_knee",
11: "left_ankle",
12: "head_top",
13: "neck"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class `mmpose.datasets.BottomUpCocoDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False)`

COCO dataset for bottom-up pose estimation.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO keypoint indexes:

```

0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- num_people: P
 - num_keypoints: K
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*list[np.ndarray(P, K, 3+tag_num)]*): Pose predictions for all people in images.
 - scores (*list[P]*): List of person scores.
 - image_path (*list[str]*): For example, `['coco/images/ val2017/000000397133.jpg']`
 - heatmap (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.BottomUpCocoWholeBodyDataset`(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

CocoWholeBodyDataset dataset for bottom-up pose estimation.

Whole-Body Human Pose Estimation in the Wild', ECCV'2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

In total, we have 133 keypoints for wholebody pose estimation.

COCO-WholeBody keypoint indexes:

```
0-16: 17 body keypoints,
17-22: 6 foot keypoints,
23-90: 68 face keypoints,
91-132: 42 hand keypoints
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class `mmpose.datasets.BottomUpCrowdPoseDataset`(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

CrowdPose dataset for bottom-up pose estimation.

“CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark”, CVPR'2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

CrowdPose keypoint indexes:

```
0: 'left_shoulder',
1: 'right_shoulder',
2: 'left_elbow',
3: 'right_elbow',
4: 'left_wrist',
5: 'right_wrist',
6: 'left_hip',
7: 'right_hip',
8: 'left_knee',
9: 'right_knee',
10: 'left_ankle',
11: 'right_ankle',
```

(continues on next page)

(continued from previous page)

```

12: 'top_head',
13: 'neck'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```

class mmpose.datasets.BottomUpMhpDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,
                                          test_mode=False)

```

MHPv2.0 dataset for top-down pose estimation.

“Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing”, ACM MM’2018. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MHP keypoint indexes:

```

0: "right ankle",
1: "right knee",
2: "right hip",
3: "left hip",
4: "left knee",
5: "left ankle",
6: "pelvis",
7: "thorax",
8: "upper neck",
9: "head top",
10: "right wrist",
11: "right elbow",
12: "right shoulder",
13: "left shoulder",
14: "left elbow",
15: "left wrist",

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class `mmpose.datasets.Compose`(*transforms*)

Compose a data pipeline with a sequence of transforms.

Parameters **transforms** (*list[dict | callable]*) – Either config dicts of transforms or transform objects.

class `mmpose.datasets.DeepFashionDataset`(*ann_file, img_prefix, data_cfg, pipeline, subset="", dataset_info=None, test_mode=False*)

DeepFashion dataset (full-body clothes) for fashion landmark detection.

“DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations”, CVPR’2016.
“Fashion Landmark Detection in the Wild”, ECCV’2016.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

The dataset contains 3 categories for full-body, upper-body and lower-body.

Fashion landmark indexes for upper-body clothes:

```
0: 'left collar',
1: 'right collar',
2: 'left sleeve',
3: 'right sleeve',
4: 'left hem',
5: 'right hem'
```

Fashion landmark indexes for lower-body clothes:

```
0: 'left waistline',
1: 'right waistline',
2: 'left hem',
3: 'right hem'
```

Fashion landmark indexes for full-body clothes:

```
0: 'left collar',
1: 'right collar',
2: 'left sleeve',
3: 'right sleeve',
4: 'left waistline',
5: 'right waistline',
6: 'left hem',
7: 'right hem'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `heatmap height`: H
 - `heatmap width`: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - `preds` (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - `boxes` (`np.ndarray[N,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
 - `image_paths` (*list[str]*): For example, [`'img_00000001.jpg'`]
 - `output_heatmap` (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **metric** (*str* | *list[str]*) – Metric to be performed. Options: `'PCK'`, `'AUC'`, `'EPE'`.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.DistributedSampler`(*dataset*, *num_replicas=None*, *rank=None*, *shuffle=True*, *seed=0*)

`DistributedSampler` inheriting from `torch.utils.data.DistributedSampler`.

In pytorch of lower versions, there is no `shuffle` argument. This child class will port one to `DistributedSampler`.

class `mmpose.datasets.Face300WDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

Face300W dataset for top-down face keypoint localization.

“300 faces In-the-wild challenge: Database and results”, Image and Vision Computing (IMAVIS) 2019.

The dataset loads raw images and apply specified transforms to return a dict containing the image tensors and other information.

The landmark annotations follow the 68 points mark-up. The definition can be found in <https://ibug.doc.ic.ac.uk/resources/300-W/>.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: `None`.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict* | *callable]*) – A sequence of data transforms.

- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='NME'*, ***kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray[1,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray[1,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - image_path (*list[str]*): For example, ['300W/ibug/ image_018.jpg']
 - output_heatmap (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'NME'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.FaceAFLWDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

Face AFLW dataset for top-down face keypoint localization.

“Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization”. In Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, 2011.

The dataset loads raw images and apply specified transforms to return a dict containing the image tensors and other information.

The landmark annotations follow the 19 points mark-up. The definition can be found in <https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/aflw/>

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.

- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='NME'*, ***kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray*[1,K,3]): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray*[1,6]): [center[0], center[1], scale[0], scale[1], area, score]
 - image_path (*list[str]*): For example, ['aflw/images/flickr/ 0/image00002.jpg']
 - output_heatmap (*np.ndarray*[N, K, H, W]): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* | *list[str]*) – Metric to be performed. Options: 'NME'.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**FaceCOFWDataset**(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

Face COFW dataset for top-down face keypoint localization.

“Robust face landmark estimation under occlusion”, ICCV’2013.

The dataset loads raw images and apply specified transforms to return a dict containing the image tensors and other information.

The landmark annotations follow the 29 points mark-up. The definition can be found in <http://www.vision.caltech.edu/xpburgos/ICCV13/>.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='NME'*, ***kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- `batch_size`: N
 - `num_keypoints`: K
 - `heatmap height`: H
 - `heatmap width`: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - `preds` (`np.ndarray[1,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - `boxes` (`np.ndarray[1,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
 - `image_path` (`list[str]`): For example, [`'cofw/images/ 000001.jpg'`]
 - `output_heatmap` (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **metric** (*str* | *list[str]*) – Metric to be performed. Options: `'NME'`.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.FaceCocoWholeBodyDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*,
dataset_info=None, *test_mode=False*)

`CocoWholeBodyDataset` for face keypoint localization.

Whole-Body Human Pose Estimation in the Wild', ECCV'2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

The face landmark annotations follow the 68 points mark-up.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: `None`.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: `False`.

evaluate(*results*, *res_folder=None*, *metric='NME'*, ***kwargs*)

Evaluate COCO-WholeBody Face keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (np.ndarray[1,K,3]): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (np.ndarray[1,6]): [center[0], center[1], scale[0], scale[1], area, score]
 - image_path (list[str]): For example, ['coco/train2017/000000000009.jpg']
 - output_heatmap (np.ndarray[N, K, H, W]): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'NME'.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**FaceWFLWDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

Face WFLW dataset for top-down face keypoint localization.

“Look at Boundary: A Boundary-Aware Face Alignment Algorithm”, CVPR’2018.

The dataset loads raw images and apply specified transforms to return a dict containing the image tensors and other information.

The landmark annotations follow the 98 points mark-up. The definition can be found in <https://wywu.github.io/projects/LAB/WFLW.html>.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results, res_folder=None, metric='NME', **kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in \${res_folder}/result_keypoints.json.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (np.ndarray[1,K,3]): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (np.ndarray[1,6]): [center[0], center[1], scale[0], scale[1], area, score]
 - image_path (list[str]): For example, ['wflw/images/0-Parade/0-Parade_marchingband_1_1015.jpg']
 - output_heatmap (np.ndarray[N, K, H, W]): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str / list[str]*) – Metric to be performed. Options: 'NME'.

Returns Evaluation results for evaluation metric.

Return type dict

class mmpose.datasets.**FreiHandDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

FreiHand dataset for top-down hand pose estimation.

“FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images”, ICCV’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

FreiHand keypoint indexes:

```
0: 'wrist',
1: 'thumb1',
2: 'thumb2',
3: 'thumb3',
4: 'thumb4',
5: 'forefinger1',
6: 'forefinger2',
7: 'forefinger3',
8: 'forefinger4',
9: 'middle_finger1',
10: 'middle_finger2',
11: 'middle_finger3',
12: 'middle_finger4',
13: 'ring_finger1',
14: 'ring_finger2',
15: 'ring_finger3',
16: 'ring_finger4',
```

(continues on next page)

(continued from previous page)

```

17: 'pinky_finger1',
18: 'pinky_finger2',
19: 'pinky_finger3',
20: 'pinky_finger4'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate freihand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['training/rgb/ 00031426.jpg']
 - **output_heatmap** (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'PCK', 'AUC', 'EPE'.

Returns Evaluation results for evaluation metric.

Return type dict

```

class mmpose.datasets.HandCocoWholeBodyDataset(ann_file, img_prefix, data_cfg, pipeline,
                                                dataset_info=None, test_mode=False)

```

CocoWholeBodyDataset for top-down hand pose estimation.

“Whole-Body Human Pose Estimation in the Wild”, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO-WholeBody Hand keypoint indexes:

```
0: 'wrist',
1: 'thumb1',
2: 'thumb2',
3: 'thumb3',
4: 'thumb4',
5: 'forefinger1',
6: 'forefinger2',
7: 'forefinger3',
8: 'forefinger4',
9: 'middle_finger1',
10: 'middle_finger2',
11: 'middle_finger3',
12: 'middle_finger4',
13: 'ring_finger1',
14: 'ring_finger2',
15: 'ring_finger3',
16: 'ring_finger4',
17: 'pinky_finger1',
18: 'pinky_finger2',
19: 'pinky_finger3',
20: 'pinky_finger4'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate COCO-WholeBody Hand keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:

- `preds` (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
- `boxes` (`np.ndarray[N,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
- `image_paths` (`list[str]`): For example, [`'Test/source/0.jpg'`]
- `output_heatmap` (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (`str`, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **metric** (`str` / `list[str]`) – Metric to be performed. Options: `'PCK'`, `'AUC'`, `'EPE'`.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.InterHand2DDataset`(`ann_file`, `camera_file`, `joint_file`, `img_prefix`, `data_cfg`, `pipeline`, `dataset_info=None`, `test_mode=False`)

InterHand2.6M 2D dataset for top-down hand pose estimation.

“InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image”, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

InterHand2.6M keypoint indexes:

```
0: 'thumb4',
1: 'thumb3',
2: 'thumb2',
3: 'thumb1',
4: 'forefinger4',
5: 'forefinger3',
6: 'forefinger2',
7: 'forefinger1',
8: 'middle_finger4',
9: 'middle_finger3',
10: 'middle_finger2',
11: 'middle_finger1',
12: 'ring_finger4',
13: 'ring_finger3',
14: 'ring_finger2',
15: 'ring_finger1',
16: 'pinky_finger4',
17: 'pinky_finger3',
18: 'pinky_finger2',
19: 'pinky_finger1',
20: 'wrist'
```

Parameters

- **ann_file** (`str`) – Path to the annotation file.
- **camera_file** (`str`) – Path to the camera file.
- **joint_file** (`str`) – Path to the joint file.

- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*str*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate interhand2d keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (`np.ndarray[N,6]`): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['Capture12/0390_dh_touchROM/cam410209/image62434.jpg']
 - output_heatmap (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'PCK', 'AUC', 'EPE'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.InterHand3DDataset`(*ann_file*, *camera_file*, *joint_file*, *img_prefix*, *data_cfg*, *pipeline*, *use_gt_root_depth=True*, *rootnet_result_file=None*, *dataset_info=None*, *test_mode=False*)

InterHand2.6M 3D dataset for top-down hand pose estimation.

“InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image”, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

InterHand2.6M keypoint indexes:

```

0: 'r_thumb4',
1: 'r_thumb3',
2: 'r_thumb2',
3: 'r_thumb1',
4: 'r_index4',
5: 'r_index3',
6: 'r_index2',
7: 'r_index1',
8: 'r_middle4',
9: 'r_middle3',
10: 'r_middle2',
11: 'r_middle1',
12: 'r_ring4',
13: 'r_ring3',
14: 'r_ring2',
15: 'r_ring1',
16: 'r_pinky4',
17: 'r_pinky3',
18: 'r_pinky2',
19: 'r_pinky1',
20: 'r_wrist',
21: 'l_thumb4',
22: 'l_thumb3',
23: 'l_thumb2',
24: 'l_thumb1',
25: 'l_index4',
26: 'l_index3',
27: 'l_index2',
28: 'l_index1',
29: 'l_middle4',
30: 'l_middle3',
31: 'l_middle2',
32: 'l_middle1',
33: 'l_ring4',
34: 'l_ring3',
35: 'l_ring2',
36: 'l_ring1',
37: 'l_pinky4',
38: 'l_pinky3',
39: 'l_pinky2',
40: 'l_pinky1',
41: 'l_wrist'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **camera_file** (*str*) – Path to the camera file.
- **joint_file** (*str*) – Path to the joint file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.

- **use_gt_root_depth** (*bool*) – Using the ground truth depth of the wrist or given depth from rootnet_result_file.
- **rootnet_result_file** (*str*) – Path to the wrist depth file.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*str*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='MPJPE'*, ***kwargs*)

Evaluate interhand2d keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **hand_type** (*np.ndarray[N, 4]*): The first two dimensions are hand type, scores is the last two dimensions.
 - **rel_root_depth** (*np.ndarray[N]*): The relative depth of left wrist and right wrist.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['Capture6/0012_aokay_upright/cam410061/image4996.jpg']
 - **output_heatmap** (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'MRRPE', 'MPJPE', 'Handedness_acc'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.MeshAdversarialDataset`(*train_dataset*, *adversarial_dataset*)

Mix Dataset for the adversarial training in 3D human mesh estimation task.

The dataset combines data from two datasets and return a dict containing data from two datasets.

Parameters

- **train_dataset** (*Dataset*) – Dataset for 3D human mesh estimation.
- **adversarial_dataset** (*Dataset*) – Dataset for adversarial learning, provides real SMPL parameters.

class `mmpose.datasets.MeshH36MDataset`(*ann_file, img_prefix, data_cfg, pipeline, test_mode=False*)

Human3.6M Dataset for 3D human mesh estimation. It inherits all function from MeshBaseDataset and has its own evaluate function.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*outputs, res_folder, metric='joint_error', logger=None*)

Evaluate 3D keypoint results.

class `mmpose.datasets.MeshMixDataset`(*configs, partition*)

Mix Dataset for 3D human mesh estimation.

The dataset combines data from multiple datasets (MeshBaseDataset) and sample the data from different datasets with the provided proportions. The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Parameters

- **configs** (*list*) – List of configs for multiple datasets.
- **partition** (*list*) – Sample proportion of multiple datasets. The length of partition should be same with that of configs. The elements of it should be non-negative and is not necessary summing up to one.

Example

```
>>> from mmpose.datasets import MeshMixDataset
>>> data_cfg = dict(
>>>     image_size=[256, 256],
>>>     iuv_size=[64, 64],
>>>     num_joints=24,
>>>     use_IUV=True,
>>>     uv_type='BF')
>>>
>>> mix_dataset = MeshMixDataset(
>>>     configs=[
>>>         dict(
>>>             ann_file='tests/data/h36m/test_h36m.npz',
>>>             img_prefix='tests/data/h36m',
>>>             data_cfg=data_cfg,
>>>             pipeline=[]),
>>>         dict(
>>>             ann_file='tests/data/h36m/test_h36m.npz',
>>>             img_prefix='tests/data/h36m',
>>>             data_cfg=data_cfg,
>>>             pipeline=[])])
```

(continues on next page)

(continued from previous page)

```
>>> ],
>>> partition=[0.6, 0.4])
```

class `mmpose.datasets.MoshDataset(ann_file, pipeline, test_mode=False)`

Mosh Dataset for the adversarial training in 3D human mesh estimation task.

The dataset return a dict containing real-world SMPL parameters.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class `mmpose.datasets.NVGestureDataset(ann_file, vid_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False)`

NVGesture dataset for gesture recognition.

“Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3D Convolutional Neural Network”, Conference on Computer Vision and Pattern Recognition (CVPR) 2016.

The dataset loads raw videos and apply specified transforms to return a dict containing the image tensors and other information.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **vid_prefix** (*str*) – Path to a directory where videos are held.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='AP'*, ***kwargs*)

Evaluate nvgesture recognition results. The gesture prediction results will be saved in `${res_folder}/result_gesture.json`.

Note:

- batch_size: N
 - heatmap length: L
-

Parameters

- **results** (*dict*) – Testing results containing the following items: - logits (*dict[str, torch.tensor[N,25,L]]*): For each item, the key represents the modality of input video, while the value represents the prediction of gesture. Three dimensions represent batch, category and temporal length, respectively. - label (*np.ndarray[N]*): [center[0], center[1], scale[0], scale[1], area, score]
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: ‘AP’.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.OneHand10KDataset`(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

OneHand10K dataset for top-down hand pose estimation.

“Mask-pose Cascaded CNN for 2D Hand Pose Estimation from Single Color Images”, TCSVT’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

OneHand10K keypoint indexes:

```
0: 'wrist',
1: 'thumb1',
2: 'thumb2',
3: 'thumb3',
4: 'thumb4',
5: 'forefinger1',
6: 'forefinger2',
7: 'forefinger3',
8: 'forefinger4',
9: 'middle_finger1',
10: 'middle_finger2',
11: 'middle_finger3',
12: 'middle_finger4',
13: 'ring_finger1',
14: 'ring_finger2',
15: 'ring_finger3',
16: 'ring_finger4',
17: 'pinky_finger1',
18: 'pinky_finger2',
19: 'pinky_finger3',
20: 'pinky_finger4'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results, res_folder=None, metric='PCK', **kwargs*)

Evaluate onehand10k keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- **batch_size**: N

- num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (`np.ndarray[N,6]`): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (`list[str]`): For example, ['Test/source/0.jpg']
 - output_heatmap (`np.ndarray[N, K, H, W]`): model outputs.
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'PCK', 'AUC', 'EPE'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.PanopticDataset`(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

Panoptic dataset for top-down hand pose estimation.

“Hand Keypoint Detection in Single Images using Multiview Bootstrapping”, CVPR’2017. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Panoptic keypoint indexes:

```
0: 'wrist',
1: 'thumb1',
2: 'thumb2',
3: 'thumb3',
4: 'thumb4',
5: 'forefinger1',
6: 'forefinger2',
7: 'forefinger3',
8: 'forefinger4',
9: 'middle_finger1',
10: 'middle_finger2',
11: 'middle_finger3',
12: 'middle_finger4',
13: 'ring_finger1',
14: 'ring_finger2',
15: 'ring_finger3',
16: 'ring_finger4',
17: 'pinky_finger1',
```

(continues on next page)

(continued from previous page)

```

18: 'pinky_finger2',
19: 'pinky_finger3',
20: 'pinky_finger4'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCKh'*, ***kwargs*)

Evaluate panoptic keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['hand_labels/ manual_test/000648952_02_1.jpg']
 - **output_heatmap** (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'PCKh', 'AUC', 'EPE'.

Returns Evaluation results for evaluation metric.

Return type dict

```

class mmpose.datasets.TopDownAicDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,
                                         test_mode=False)

```

AicDataset dataset for top-down pose estimation.

“AI Challenger : A Large-scale Dataset for Going Deeper in Image Understanding”, arXiv’2017. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

AIC keypoint indexes:

```
0: "right_shoulder",
1: "right_elbow",
2: "right_wrist",
3: "left_shoulder",
4: "left_elbow",
5: "left_wrist",
6: "right_hip",
7: "right_knee",
8: "right_ankle",
9: "left_hip",
10: "left_knee",
11: "left_ankle",
12: "head_top",
13: "neck"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class `mmpose.datasets.TopDownCocoDataset`(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

CocoDataset dataset for top-down pose estimation.

“Microsoft COCO: Common Objects in Context”, ECCV’2014. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO keypoint indexes:

```
0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
```

(continues on next page)

(continued from previous page)

```

10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['data/coco/val2017/0000000393226.jpg']
 - **heatmap** (*np.ndarray[N, K, H, W]*): model output heatmap
 - **bbox_id** (*list(int)*).
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.TopDownCocoWholeBodyDataset(ann_file, img_prefix, data_cfg, pipeline,  
                                                  dataset_info=None, test_mode=False)
```

CocoWholeBodyDataset dataset for top-down pose estimation.

“Whole-Body Human Pose Estimation in the Wild”, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO-WholeBody keypoint indexes:

```
0-16: 17 body keypoints,  
17-22: 6 foot keypoints,  
23-90: 68 face keypoints,  
91-132: 42 hand keypoints
```

In total, we have 133 keypoints for wholebody pose estimation.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.TopDownCrowdPoseDataset(ann_file, img_prefix, data_cfg, pipeline,  
                                              dataset_info=None, test_mode=False)
```

CrowdPoseDataset dataset for top-down pose estimation.

“CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark”, CVPR’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

CrowdPose keypoint indexes:

```
0: 'left_shoulder',  
1: 'right_shoulder',  
2: 'left_elbow',  
3: 'right_elbow',  
4: 'left_wrist',  
5: 'right_wrist',  
6: 'left_hip',  
7: 'right_hip',  
8: 'left_knee',  
9: 'right_knee',  
10: 'left_ankle',  
11: 'right_ankle',  
12: 'top_head',  
13: 'neck'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class mmpose.datasets.**TopDownFreiHandDataset**(*args, **kwargs)

Deprecated TopDownFreiHandDataset.

evaluate(*cfg, preds, output_dir, *args, **kwargs*)

Evaluate keypoint results.

class mmpose.datasets.**TopDownH36MDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

Human3.6M dataset for top-down 2D pose estimation.

“Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”, TPAMI 2014. More details can be found in the [paper](#).

Human3.6M keypoint indexes:

```
0: 'root (pelvis)',
1: 'right_hip',
2: 'right_knee',
3: 'right_foot',
4: 'left_hip',
5: 'left_knee',
6: 'left_foot',
7: 'spine',
8: 'thorax',
9: 'neck_base',
10: 'head',
11: 'left_shoulder',
12: 'left_elbow',
13: 'left_wrist',
14: 'right_shoulder',
15: 'right_elbow',
16: 'right_wrist'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate human3.6m 2d keypoint results. The pose prediction results will be saved in `$(res_folder)/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (`np.ndarray[N,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
 - **image_paths** (`list[str]`): For example, [`'data/coco/val2017/000000393226.jpg'`]
 - **heatmap** (`np.ndarray[N, K, H, W]`): model output heatmap
 - **bbox_id** (`list(int)`).
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* | *list[str]*) – Metric to be performed. Defaults: 'PCK'.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.TopDownJhmdbDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

JhmdbDataset dataset for top-down pose estimation.

“Towards understanding action recognition”, ICCV’2013. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

sub-JHMDB keypoint indexes:

```
0: "neck",
1: "belly",
2: "head",
3: "right_shoulder",
4: "left_shoulder",
5: "right_hip",
6: "left_hip",
7: "right_elbow",
8: "left_elbow",
9: "right_knee",
```

(continues on next page)

(continued from previous page)

```

10: "left_knee",
11: "right_wrist",
12: "left_wrist",
13: "right_ankle",
14: "left_ankle"

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate onehand10k keypoint results. The pose prediction results will be saved in `#{res_folder}/result_keypoints.json`.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_path** (*list[str]*)
 - **output_heatmap** (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Options: 'PCK', 'tPCK'. PCK means normalized by the bounding boxes, while tPCK means normalized by the torso size.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.TopDownMhpDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,
                                         test_mode=False)
```

MHPv2.0 dataset for top-down pose estimation.

“Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing”, ACM MM’2018. More details can be found in the [paper](#)

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. ‘<https://github.com/ZhaoJ9014/Multi-Human-Parsing/tree/master/>’ ‘Evaluation/Multi-Human-Pose’ Please be cautious if you use the results in papers.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MHP keypoint indexes:

```
0: "right ankle",
1: "right knee",
2: "right hip",
3: "left hip",
4: "left knee",
5: "left ankle",
6: "pelvis",
7: "thorax",
8: "upper neck",
9: "head top",
10: "right wrist",
11: "right elbow",
12: "right shoulder",
13: "left shoulder",
14: "left elbow",
15: "left wrist",
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.TopDownMpiiDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None,
                                         test_mode=False)
```

MPII Dataset for top-down pose estimation.

“2D Human Pose Estimation: New Benchmark and State of the Art Analysis” ,CVPR’2014. More details can be found in the [paper](#) .

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MPII keypoint indexes:


```

0: 'right_ankle'
1: 'right_knee',
2: 'right_hip',
3: 'left_hip',
4: 'left_knee',
5: 'left_ankle',
6: 'pelvis',
7: 'thorax',
8: 'upper_neck',
9: 'head_top',
10: 'right_wrist',
11: 'right_elbow',
12: 'right_shoulder',
13: 'left_shoulder',
14: 'left_elbow',
15: 'left_wrist'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCKh'*, ***kwargs*)

Evaluate PCKh for MPII dataset. Adapted from <https://github.com/leoxiaobin/deep-high-resolution-net.pytorch> Copyright (c) Microsoft, under the MIT License.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['val2017/000000 397133.jpg']
 - heatmap (*np.ndarray[N, K, H, W]*): model output heatmap.

- **res_folder** (*str*, *optional*) – The folder to save the testing results. Default: `None`.
- **metric** (*str* / *list[str]*) – Metrics to be performed. Defaults: `'PCKh'`.

Returns PCKh for each joint

Return type dict

class `mmpose.datasets.TopDownMpiiTrbDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*,
dataset_info=None, *test_mode=False*)

MPII-TRB Dataset dataset for top-down pose estimation.

“TRB: A Novel Triplet Representation for Understanding 2D Human Body”, ICCV’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MPII-TRB keypoint indexes:

```
0: 'left_shoulder'
1: 'right_shoulder'
2: 'left_elbow'
3: 'right_elbow'
4: 'left_wrist'
5: 'right_wrist'
6: 'left_hip'
7: 'right_hip'
8: 'left_knee'
9: 'right_knee'
10: 'left_ankle'
11: 'right_ankle'
12: 'head'
13: 'neck'

14: 'right_neck'
15: 'left_neck'
16: 'medial_right_shoulder'
17: 'lateral_right_shoulder'
18: 'medial_right_bow'
19: 'lateral_right_bow'
20: 'medial_right_wrist'
21: 'lateral_right_wrist'
22: 'medial_left_shoulder'
23: 'lateral_left_shoulder'
24: 'medial_left_bow'
25: 'lateral_left_bow'
26: 'medial_left_wrist'
27: 'lateral_left_wrist'
28: 'medial_right_hip'
29: 'lateral_right_hip'
30: 'medial_right_knee'
31: 'lateral_right_knee'
32: 'medial_right_ankle'
33: 'lateral_right_ankle'
```

(continues on next page)

(continued from previous page)

```

34: 'medial_left_hip'
35: 'lateral_left_hip'
36: 'medial_left_knee'
37: 'lateral_left_knee'
38: 'medial_left_ankle'
39: 'lateral_left_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCKh'*, ***kwargs*)

Evaluate PCKh for MPII-TRB dataset.

Note:

- batch_size: N
- num_keypoints: K
- heatmap height: H
- heatmap width: W

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['/val2017/ 000000397133.jpg']
 - **heatmap** (*np.ndarray[N, K, H, W]*): model output heatmap.
 - **bbox_ids** (*list[str]*): For example, ['27407'].
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metrics to be performed. Defaults: 'PCKh'.

Returns PCKh for each joint

Return type dict

```
class mmpose.datasets.TopDownOCHumanDataset(ann_file, img_prefix, data_cfg, pipeline,  
                                             dataset_info=None, test_mode=False)
```

OCHuman dataset for top-down pose estimation.

“Pose2Seg: Detection Free Human Instance Segmentation”, CVPR’2019. More details can be found in the [paper](#).

“Occluded Human (OCHuman)” dataset contains 8110 heavily occluded human instances within 4731 images. OCHuman dataset is designed for validation and testing. To evaluate on OCHuman, the model should be trained on COCO training set, and then test the robustness of the model to occlusion using OCHuman.

OCHuman keypoint indexes (same as COCO):

```
0: 'nose',  
1: 'left_eye',  
2: 'right_eye',  
3: 'left_ear',  
4: 'right_ear',  
5: 'left_shoulder',  
6: 'right_shoulder',  
7: 'left_elbow',  
8: 'right_elbow',  
9: 'left_wrist',  
10: 'right_wrist',  
11: 'left_hip',  
12: 'right_hip',  
13: 'left_knee',  
14: 'right_knee',  
15: 'left_ankle',  
16: 'right_ankle'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.TopDownOneHand10KDataset(*args, **kwargs)
```

Deprecated TopDownOneHand10KDataset.

```
evaluate(cfg, preds, output_dir, *args, **kwargs)
```

Evaluate keypoint results.

```
class mmpose.datasets.TopDownPanopticDataset(*args, **kwargs)
```

Deprecated TopDownPanopticDataset.

```
evaluate(cfg, preds, output_dir, *args, **kwargs)
```

Evaluate keypoint results.

```
class mmpose.datasets.TopDownPoseTrack18Dataset(ann_file, img_prefix, data_cfg, pipeline,  
                                                  dataset_info=None, test_mode=False)
```

PoseTrack18 dataset for top-down pose estimation.

“Posetrack: A benchmark for human pose estimation and tracking”, CVPR’2018. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

PoseTrack2018 keypoint indexes:

```
0: 'nose',
1: 'head_bottom',
2: 'head_top',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate posetrack keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- num_keypoints: K
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['val/010016_mpii_test /000024.jpg']

- heatmap (np.ndarray[N, K, H, W]): model output heatmap.
- bbox_id (list(int))
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* | *list[str]*) – Metric to be performed. Defaults: ‘mAP’.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.TopDownPoseTrack18VideoDataset(ann_file, img_prefix, data_cfg, pipeline,
                                                    dataset_info=None, test_mode=False,
                                                    ph_fill_len=6)
```

PoseTrack18 dataset for top-down pose estimation.

“Posetrack: A benchmark for human pose estimation and tracking”, CVPR’2018. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

PoseTrack2018 keypoint indexes:

```
0: 'nose',
1: 'head_bottom',
2: 'head_top',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where videos/images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.
- **ph_fill_len** (*int*) – The length of the placeholder to fill in the image filenames, default: 6 in PoseTrack18.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate posetrack keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- `num_keypoints: K`
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - `preds` (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - `boxes` (`np.ndarray[N,6]`): `[center[0], center[1], scale[0], scale[1], area, score]`
 - `image_paths` (`list[str]`): For example, `['val/010016_mpii_test /000024.jpg']`
 - `heatmap` (`np.ndarray[N, K, H, W]`): model output heatmap.
 - `bbox_id` (`list(int)`)
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: `'mAP'`.

Returns Evaluation results for evaluation metric.

Return type dict

`mmpose.datasets.build_data_loader(dataset, samples_per_gpu, workers_per_gpu, num_gpus=1, dist=True, shuffle=True, seed=None, drop_last=True, pin_memory=True, **kwargs)`

Build PyTorch DataLoader.

In distributed training, each GPU/process has a dataloader. In non-distributed training, there is only one dataloader for all GPUs.

Parameters

- **dataset** (*Dataset*) – A PyTorch dataset.
- **samples_per_gpu** (*int*) – Number of training samples on each GPU, i.e., batch size of each GPU.
- **workers_per_gpu** (*int*) – How many subprocesses to use for data loading for each GPU.
- **num_gpus** (*int*) – Number of GPUs. Only used in non-distributed training.
- **dist** (*bool*) – Distributed training/test or not. Default: `True`.
- **shuffle** (*bool*) – Whether to shuffle the data at every epoch. Default: `True`.
- **drop_last** (*bool*) – Whether to drop the last incomplete batch in epoch. Default: `True`
- **pin_memory** (*bool*) – Whether to use `pin_memory` in `DataLoader`. Default: `True`
- **kwargs** – any keyword argument to be used to initialize `DataLoader`

Returns A PyTorch dataloader.

Return type `DataLoader`

`mmpose.datasets.build_dataset(cfg, default_args=None)`

Build a dataset from config dict.

Parameters

- **cfg** (*dict*) – Config dict. It should at least contain the key “type”.
- **default_args** (*dict*, *optional*) – Default initialization arguments. Default: None.

Returns The constructed dataset.

Return type Dataset

50.1 datasets

`class mmpose.datasets.datasets.top_down.TopDownAicDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False)`

AicDataset dataset for top-down pose estimation.

“AI Challenger : A Large-scale Dataset for Going Deeper in Image Understanding”, arXiv’2017. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

AIC keypoint indexes:

```
0: "right_shoulder",
1: "right_elbow",
2: "right_wrist",
3: "left_shoulder",
4: "left_elbow",
5: "left_wrist",
6: "right_hip",
7: "right_knee",
8: "right_ankle",
9: "left_hip",
10: "left_knee",
11: "left_ankle",
12: "head_top",
13: "neck"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

`class mmpose.datasets.datasets.top_down.TopDownCocoDataset(ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False)`

CocoDataset dataset for top-down pose estimation.

“Microsoft COCO: Common Objects in Context”, ECCV’2014. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO keypoint indexes:

```
0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.

- `boxes` (`np.ndarray[N,6]`): [`center[0]`, `center[1]`, `scale[0]`, `scale[1]`, `area`, `score`]
- `image_paths` (`list[str]`): For example, [`'data/coco/val2017/000000393226.jpg'`]
- `heatmap` (`np.ndarray[N, K, H, W]`): model output heatmap
- `bbox_id` (`list(int)`).
- **`res_folder`** (`str`, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: `None`.
- **`metric`** (`str` | `list[str]`) – Metric to be performed. Defaults: `'mAP'`.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.datasets.top_down.TopDownCocoWholeBodyDataset(ann_file, img_prefix,
                                                                    data_cfg, pipeline,
                                                                    dataset_info=None,
                                                                    test_mode=False)
```

CocoWholeBodyDataset dataset for top-down pose estimation.

“Whole-Body Human Pose Estimation in the Wild”, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO-WholeBody keypoint indexes:

```
0-16: 17 body keypoints,
17-22: 6 foot keypoints,
23-90: 68 face keypoints,
91-132: 42 hand keypoints
```

In total, we have 133 keypoints for wholebody pose estimation.

Parameters

- **`ann_file`** (`str`) – Path to the annotation file.
- **`img_prefix`** (`str`) – Path to a directory where images are held. Default: `None`.
- **`data_cfg`** (`dict`) – config
- **`pipeline`** (`list[dict | callable]`) – A sequence of data transforms.
- **`dataset_info`** (`DatasetInfo`) – A class containing all dataset info.
- **`test_mode`** (`bool`) – Store True when building test or validation dataset. Default: `False`.

```
class mmpose.datasets.datasets.top_down.TopDownCrowdPoseDataset(ann_file, img_prefix, data_cfg,
                                                                pipeline, dataset_info=None,
                                                                test_mode=False)
```

CrowdPoseDataset dataset for top-down pose estimation.

“CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark”, CVPR’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

CrowdPose keypoint indexes:

```

0: 'left_shoulder',
1: 'right_shoulder',
2: 'left_elbow',
3: 'right_elbow',
4: 'left_wrist',
5: 'right_wrist',
6: 'left_hip',
7: 'right_hip',
8: 'left_knee',
9: 'right_knee',
10: 'left_ankle',
11: 'right_ankle',
12: 'top_head',
13: 'neck'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class mmpose.datasets.datasets.top_down.**TopDownH36MDataset**(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

Human3.6M dataset for top-down 2D pose estimation.

“Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”, TPAMI 2014. More details can be found in the [paper](#).

Human3.6M keypoint indexes:

```

0: 'root (pelvis)',
1: 'right_hip',
2: 'right_knee',
3: 'right_foot',
4: 'left_hip',
5: 'left_knee',
6: 'left_foot',
7: 'spine',
8: 'thorax',
9: 'neck_base',
10: 'head',
11: 'left_shoulder',
12: 'left_elbow',
13: 'left_wrist',
14: 'right_shoulder',
15: 'right_elbow',
16: 'right_wrist'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate human3.6m 2d keypoint results. The pose prediction results will be saved in `$(res_folder)/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (`np.ndarray[N,K,3]`): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (`np.ndarray[N,6]`): `[center[0], center[1], scale[0], scale[1], area, score]`
 - **image_paths** (*list[str]*): For example, `['data/coco/val2017/000000393226.jpg']`
 - **heatmap** (`np.ndarray[N, K, H, W]`): model output heatmap
 - **bbox_id** (*list(int)*).
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'PCK'.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.datasets.top_down.TopDownHalpeDataset(ann_file, img_prefix, data_cfg,  
                                                         pipeline, dataset_info=None,  
                                                         test_mode=False)
```

HalpeDataset for top-down pose estimation.

`'https://github.com/Fang-Haosu/Halpe-FullBody'`

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Halpe keypoint indexes:

```

0-19: 20 body keypoints,
20-25: 6 foot keypoints,
26-93: 68 face keypoints,
94-135: 42 hand keypoints

```

In total, we have 136 keypoints **for** wholebody pose estimation.

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```

class mmpose.datasets.datasets.top_down.TopDownJhmdbDataset(ann_file, img_prefix, data_cfg,
                                                            pipeline, dataset_info=None,
                                                            test_mode=False)

```

JhmdbDataset dataset for top-down pose estimation.

“Towards understanding action recognition”, ICCV’2013. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

sub-JHMDB keypoint indexes:

```

0: "neck",
1: "belly",
2: "head",
3: "right_shoulder",
4: "left_shoulder",
5: "right_hip",
6: "left_hip",
7: "right_elbow",
8: "left_elbow",
9: "right_knee",
10: "left_knee",
11: "right_wrist",
12: "left_wrist",
13: "right_ankle",
14: "left_ankle"

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.

- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCK'*, ***kwargs*)

Evaluate onehand10k keypoint results. The pose prediction results will be saved in `$(res_folder)/result_keypoints.json`.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - image_path (*list[str]*)
 - output_heatmap (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* | *list[str]*) – Metric to be performed. Options: 'PCK', 'tPCK'. PCK means normalized by the bounding boxes, while tPCK means normalized by the torso size.

Returns Evaluation results for evaluation metric.

Return type dict

class `mmpose.datasets.datasets.top_down.TopDownMhpDataset`(*ann_file*, *img_prefix*, *data_cfg*, *pipeline*, *dataset_info=None*, *test_mode=False*)

MHPv2.0 dataset for top-down pose estimation.

“Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing”, ACM MM’2018. More details can be found in the [paper](#)

Note that, the evaluation metric used here is mAP (adapted from COCO), which may be different from the official evaluation codes. ‘<https://github.com/ZhaoJ9014/Multi-Human-Parsing/tree/master/>’ ‘Evaluation/Multi-Human-Pose’ Please be cautious if you use the results in papers.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MHP keypoint indexes:

```
0: "right ankle",
1: "right knee",
2: "right hip",
3: "left hip",
4: "left knee",
```

(continues on next page)

(continued from previous page)

```

5: "left ankle",
6: "pelvis",
7: "thorax",
8: "upper neck",
9: "head top",
10: "right wrist",
11: "right elbow",
12: "right shoulder",
13: "left shoulder",
14: "left elbow",
15: "left wrist",

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

class mmpose.datasets.datasets.top_down.TopDownMpiiDataset(*ann_file, img_prefix, data_cfg, pipeline, dataset_info=None, test_mode=False*)

MPII Dataset for top-down pose estimation.

“2D Human Pose Estimation: New Benchmark and State of the Art Analysis” ,CVPR’2014. More details can be found in the [paper](#) .

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MPII keypoint indexes:

```

0: 'right_ankle'
1: 'right_knee',
2: 'right_hip',
3: 'left_hip',
4: 'left_knee',
5: 'left_ankle',
6: 'pelvis',
7: 'thorax',
8: 'upper_neck',
9: 'head_top',
10: 'right_wrist',
11: 'right_elbow',
12: 'right_shoulder',
13: 'left_shoulder',
14: 'left_elbow',
15: 'left_wrist'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate (*results*, *res_folder=None*, *metric='PCKh'*, ***kwargs*)

Evaluate PCKh for Mpii dataset. Adapted from <https://github.com/leoxiaobin/deep-high-resolution-net.pytorch> Copyright (c) Microsoft, under the MIT License.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (*list[str]*): For example, ['val2017/000000 397133.jpg']
 - heatmap (*np.ndarray[N, K, H, W]*): model output heatmap.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. Default: None.
- **metric** (*str | list[str]*) – Metrics to be performed. Defaults: 'PCKh'.

Returns PCKh for each joint

Return type dict

```
class mmpose.datasets.datasets.top_down.TopDownMpiiTrbDataset(ann_file, img_prefix, data_cfg,  
                                                             pipeline, dataset_info=None,  
                                                             test_mode=False)
```

Mpii-TRB Dataset dataset for top-down pose estimation.

“TRB: A Novel Triplet Representation for Understanding 2D Human Body”, ICCV’2019. More details can be found in the [paper](#) .

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

Mpii-TRB keypoint indexes:


```

0: 'left_shoulder'
1: 'right_shoulder'
2: 'left_elbow'
3: 'right_elbow'
4: 'left_wrist'
5: 'right_wrist'
6: 'left_hip'
7: 'right_hip'
8: 'left_knee'
9: 'right_knee'
10: 'left_ankle'
11: 'right_ankle'
12: 'head'
13: 'neck'

14: 'right_neck'
15: 'left_neck'
16: 'medial_right_shoulder'
17: 'lateral_right_shoulder'
18: 'medial_right_bow'
19: 'lateral_right_bow'
20: 'medial_right_wrist'
21: 'lateral_right_wrist'
22: 'medial_left_shoulder'
23: 'lateral_left_shoulder'
24: 'medial_left_bow'
25: 'lateral_left_bow'
26: 'medial_left_wrist'
27: 'lateral_left_wrist'
28: 'medial_right_hip'
29: 'lateral_right_hip'
30: 'medial_right_knee'
31: 'lateral_right_knee'
32: 'medial_right_ankle'
33: 'lateral_right_ankle'
34: 'medial_left_hip'
35: 'lateral_left_hip'
36: 'medial_left_knee'
37: 'lateral_left_knee'
38: 'medial_left_ankle'
39: 'lateral_left_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='PCKh'*, ***kwargs*)
Evaluate PCKh for MPII-TRB dataset.

Note:

- batch_size: N
 - num_keypoints: K
 - heatmap height: H
 - heatmap width: W
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (np.ndarray[N,K,3]): The first two dimensions are coordinates, score is the third dimension of the array.
 - boxes (np.ndarray[N,6]): [center[0], center[1], scale[0], scale[1], area, score]
 - image_paths (list[str]): For example, ['/val2017/ 000000397133.jpg']
 - heatmap (np.ndarray[N, K, H, W]): model output heatmap.
 - bbox_ids (list[str]): For example, ['27407'].
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str* / *list[str]*) – Metrics to be performed. Defaults: 'PCKh'.

Returns PCKh for each joint

Return type dict

class mmpose.datasets.datasets.top_down.**TopDownOCHumanDataset**(*ann_file*, *img_prefix*, *data_cfg*,
pipeline, *dataset_info=None*,
test_mode=False)

OCHuman dataset for top-down pose estimation.

“Pose2Seg: Detection Free Human Instance Segmentation”, CVPR’2019. More details can be found in the [paper](#).

“Occluded Human (OCHuman)” dataset contains 8110 heavily occluded human instances within 4731 images. OCHuman dataset is designed for validation and testing. To evaluate on OCHuman, the model should be trained on COCO training set, and then test the robustness of the model to occlusion using OCHuman.

OCHuman keypoint indexes (same as COCO):

```
0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
```

(continues on next page)

(continued from previous page)

```

10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```

class mmpose.datasets.datasets.top_down.TopDownPoseTrack18Dataset(ann_file, img_prefix, data_cfg,
                                                                    pipeline, dataset_info=None,
                                                                    test_mode=False)

```

PoseTrack18 dataset for top-down pose estimation.

“Posetrack: A benchmark for human pose estimation and tracking”, CVPR’2018. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

PoseTrack2018 keypoint indexes:

```

0: 'nose',
1: 'head_bottom',
2: 'head_top',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.

- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate (*results, res_folder=None, metric='mAP', **kwargs*)

Evaluate posetrack keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- num_keypoints: K
-

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['val/010016_mpii_test /000024.jpg']
 - **heatmap** (*np.ndarray[N, K, H, W]*): model output heatmap.
 - **bbox_id** (*list(int)*)
- **res_folder** (*str, optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.datasets.top_down.TopDownPoseTrack18VideoDataset(ann_file, img_prefix,
                                                                    data_cfg, pipeline,
                                                                    dataset_info=None,
                                                                    test_mode=False,
                                                                    ph_fill_len=6)
```

PoseTrack18 dataset for top-down pose estimation.

“Posetrack: A benchmark for human pose estimation and tracking”, CVPR’2018. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

PoseTrack2018 keypoint indexes:

```
0: 'nose',
1: 'head_bottom',
2: 'head_top',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
```

(continues on next page)

(continued from previous page)

```

6: 'right_shoulder',
7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where videos/images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.
- **ph_fill_len** (*int*) – The length of the placeholder to fill in the image filenames, default: 6 in PoseTrack18.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate posetrack keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- num_keypoints: K

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - **preds** (*np.ndarray[N,K,3]*): The first two dimensions are coordinates, score is the third dimension of the array.
 - **boxes** (*np.ndarray[N,6]*): [center[0], center[1], scale[0], scale[1], area, score]
 - **image_paths** (*list[str]*): For example, ['val/010016_mpii_test /000024.jpg']
 - **heatmap** (*np.ndarray[N, K, H, W]*): model output heatmap.
 - **bbox_id** (*list(int)*)
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.datasets.bottom_up.BottomUpAicDataset(ann_file, img_prefix, data_cfg,
                                                            pipeline, dataset_info=None,
                                                            test_mode=False)
```

Aic dataset for bottom-up pose estimation.

“AI Challenger : A Large-scale Dataset for Going Deeper in Image Understanding”, arXiv’2017. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

AIC keypoint indexes:

```
0: "right_shoulder",
1: "right_elbow",
2: "right_wrist",
3: "left_shoulder",
4: "left_elbow",
5: "left_wrist",
6: "right_hip",
7: "right_knee",
8: "right_ankle",
9: "left_hip",
10: "left_knee",
11: "left_ankle",
12: "head_top",
13: "neck"
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.datasets.bottom_up.BottomUpCocoDataset(ann_file, img_prefix, data_cfg,
                                                             pipeline, dataset_info=None,
                                                             test_mode=False)
```

COCO dataset for bottom-up pose estimation.

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

COCO keypoint indexes:

```
0: 'nose',
1: 'left_eye',
2: 'right_eye',
3: 'left_ear',
4: 'right_ear',
5: 'left_shoulder',
6: 'right_shoulder',
```

(continues on next page)

(continued from previous page)

```

7: 'left_elbow',
8: 'right_elbow',
9: 'left_wrist',
10: 'right_wrist',
11: 'left_hip',
12: 'right_hip',
13: 'left_knee',
14: 'right_knee',
15: 'left_ankle',
16: 'right_ankle'

```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

evaluate(*results*, *res_folder=None*, *metric='mAP'*, ***kwargs*)

Evaluate coco keypoint results. The pose prediction results will be saved in `${res_folder}/result_keypoints.json`.

Note:

- num_people: P
- num_keypoints: K

Parameters

- **results** (*list[dict]*) – Testing results containing the following items:
 - preds (*list[np.ndarray(P, K, 3+tag_num)]*): Pose predictions for all people in images.
 - scores (*list[P]*): List of person scores.
 - image_path (*list[str]*): For example, [`'coco/images/ val2017/000000397133.jpg'`]
 - heatmap (*np.ndarray[N, K, H, W]*): model outputs.
- **res_folder** (*str*, *optional*) – The folder to save the testing results. If not specified, a temp folder will be created. Default: None.
- **metric** (*str | list[str]*) – Metric to be performed. Defaults: 'mAP'.

Returns Evaluation results for evaluation metric.

Return type dict

```
class mmpose.datasets.datasets.bottom_up.BottomUpCocoWholeBodyDataset(ann_file, img_prefix,
                                                                    data_cfg, pipeline,
                                                                    dataset_info=None,
                                                                    test_mode=False)
```

CocoWholeBodyDataset dataset for bottom-up pose estimation.

Whole-Body Human Pose Estimation in the Wild’, ECCV’2020. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

In total, we have 133 keypoints for wholebody pose estimation.

COCO-WholeBody keypoint indexes:

```
0-16: 17 body keypoints,
17-22: 6 foot keypoints,
23-90: 68 face keypoints,
91-132: 42 hand keypoints
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.datasets.bottom_up.BottomUpCrowdPoseDataset(ann_file, img_prefix, data_cfg,
                                                                    pipeline, dataset_info=None,
                                                                    test_mode=False)
```

CrowdPose dataset for bottom-up pose estimation.

“CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark”, CVPR’2019. More details can be found in the [paper](#).

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

CrowdPose keypoint indexes:

```
0: 'left_shoulder',
1: 'right_shoulder',
2: 'left_elbow',
3: 'right_elbow',
4: 'left_wrist',
5: 'right_wrist',
6: 'left_hip',
7: 'right_hip',
8: 'left_knee',
9: 'right_knee',
10: 'left_ankle',
11: 'right_ankle',
```

(continues on next page)

(continued from previous page)

```
12: 'top_head',
13: 'neck'
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

```
class mmpose.datasets.datasets.bottom_up.BottomUpMhpDataset(ann_file, img_prefix, data_cfg,
                                                            pipeline, dataset_info=None,
                                                            test_mode=False)
```

MHPv2.0 dataset for top-down pose estimation.

“Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing”, ACM MM’2018. More details can be found in the [paper](#)

The dataset loads raw features and apply specified transforms to return a dict containing the image tensors and other information.

MHP keypoint indexes:

```
0: "right ankle",
1: "right knee",
2: "right hip",
3: "left hip",
4: "left knee",
5: "left ankle",
6: "pelvis",
7: "thorax",
8: "upper neck",
9: "head top",
10: "right wrist",
11: "right elbow",
12: "right shoulder",
13: "left shoulder",
14: "left elbow",
15: "left wrist",
```

Parameters

- **ann_file** (*str*) – Path to the annotation file.
- **img_prefix** (*str*) – Path to a directory where images are held. Default: None.
- **data_cfg** (*dict*) – config
- **pipeline** (*list[dict | callable]*) – A sequence of data transforms.
- **dataset_info** (*DatasetInfo*) – A class containing all dataset info.
- **test_mode** (*bool*) – Store True when building test or validation dataset. Default: False.

50.2 pipelines

```
class mmpose.datasets.pipelines.loading.LoadImageFromFile(to_float32=False, color_type='color',
                                                         channel_order='rgb',
                                                         file_client_args={'backend': 'disk'})
```

Loading image(s) from file.

Required key: “image_file”.

Added key: “img”.

Parameters

- **to_float32** (*bool*) – Whether to convert the loaded image to a float32 numpy array. If set to False, the loaded image is an uint8 array. Defaults to False.
- **color_type** (*str*) – Flags specifying the color type of a loaded image, candidates are ‘color’, ‘grayscale’ and ‘unchanged’.
- **channel_order** (*str*) – Order of channel, candidates are ‘bgr’ and ‘rgb’.
- **file_client_args** (*dict*) – Arguments to instantiate a FileClient. See `mmpcv.fileio.FileClient` for details. Defaults to `dict(backend='disk')`.

```
class mmpose.datasets.pipelines.loading.LoadVideoFromFile(to_float32=False,
                                                         file_client_args={'backend': 'disk'})
```

Loading video(s) from file.

Required key: “video_file”.

Added key: “video”.

Parameters

- **to_float32** (*bool*) – Whether to convert the loaded video to a float32 numpy array. If set to False, the loaded video is an uint8 array. Defaults to False.
- **file_client_args** (*dict*) – Arguments to instantiate a FileClient. See `mmpcv.fileio.FileClient` for details. Defaults to `dict(backend='disk')`.

```
class mmpose.datasets.pipelines.shared_transform.Albumentation(transforms, keymap=None)
```

Albumentation augmentation (pixel-level transforms only). Adds custom pixel-level transformations from Albumentations library. Please visit <https://albumentations.readthedocs.io> to get more information.

Note: we only support pixel-level transforms. Please visit <https://github.com/albumentations-team/albumentations#pixel-level-transforms> to get more information about pixel-level transforms.

An example of transforms is as followed:

```
[
    dict(
        type='RandomBrightnessContrast',
        brightness_limit=[0.1, 0.3],
        contrast_limit=[0.1, 0.3],
        p=0.2),
    dict(type='ChannelShuffle', p=0.1),
    dict(
        type='OneOf',
        transforms=[
            dict(type='Blur', blur_limit=3, p=1.0),
            dict(type='MedianBlur', blur_limit=3, p=1.0)
```

(continues on next page)

(continued from previous page)

```

    ],
    p=0.1),
]

```

Parameters

- **transforms** (*list[dict]*) – A list of Albumentation transformations
- **keymap** (*dict*) – Contains {'input key': 'albumentation-style key'}, e.g., {'img': 'image'}.

albu_builder(*cfg*)

Import a module from albumentations.

It resembles some of `build_from_cfg()` logic.

Parameters **cfg** (*dict*) – Config dict. It should at least contain the key “type”.

Returns The constructed object.

Return type `obj`

static mapper(*d, keymap*)

Dictionary mapper.

Renames keys according to keymap provided.

Parameters

- **d** (*dict*) – old dict
- **keymap** (*dict*) – {'old_key': 'new_key'}

Returns new dict.

Return type `dict`

```

class mmpose.datasets.pipelines.shared_transform.Collect(keys, meta_keys,
                                                         meta_name='img metas')

```

Collect data from the loader relevant to the specific task.

This keeps the items in *keys* as it is, and collect items in *meta_keys* into a meta item called *meta_name*. This is usually the last stage of the data loader pipeline. For example, when *keys*=‘imgs’, *meta_keys*=('filename', 'label', 'original_shape'), *meta_name*=‘img metas’, the results will be a dict with keys ‘imgs’ and ‘img metas’, where ‘img metas’ is a DataContainer of another dict with keys ‘filename’, ‘label’, ‘original_shape’.

Parameters

- **keys** (*Sequence[str|tuple]*) – Required keys to be collected. If a tuple (key, key_new) is given as an element, the item retrieved by key will be renamed as key_new in collected data.
- **meta_name** (*str*) – The name of the key that contains meta information. This key is always populated. Default: “img metas”.
- **meta_keys** (*Sequence[str|tuple]*) – Keys that are collected under meta_name. The contents of the *meta_name* dictionary depends on *meta_keys*.

```

class mmpose.datasets.pipelines.shared_transform.Compose(transforms)

```

Compose a data pipeline with a sequence of transforms.

Parameters **transforms** (*list[dict | callable]*) – Either config dicts of transforms or transform objects.

class `mmpose.datasets.pipelines.shared_transform.MultiItemProcess`(*pipeline*)

Process each item and merge multi-item results to lists.

Parameters `pipeline` (*dict*) – Dictionary to construct pipeline for a single item.

class `mmpose.datasets.pipelines.shared_transform.MultitaskGatherTarget`(*pipeline_list*,
pipeline_indices=None,
keys=('target',
'target_weight'))

Gather the targets for multitask heads.

Parameters

- **pipeline_list** (*list[list]*) – List of pipelines for all heads.
- **pipeline_indices** (*list[int]*) – Pipeline index of each head.

class `mmpose.datasets.pipelines.shared_transform.NormalizeTensor`(*mean*, *std*)

Normalize the Tensor image (CxHxW), with mean and std.

Required key: 'img'. Modifies key: 'img'.

Parameters

- **mean** (*list[float]*) – Mean values of 3 channels.
- **std** (*list[float]*) – Std values of 3 channels.

class `mmpose.datasets.pipelines.shared_transform.PhotometricDistortion`(*brightness_delta=32*,
contrast_range=(0.5,
1.5),
saturation_range=(0.5,
1.5), *hue_delta=18*)

Apply photometric distortion to image sequentially, every transformation is applied with a probability of 0.5. The position of random contrast is in second or second to last.

1. random brightness
2. random contrast (mode 0)
3. convert color from BGR to HSV
4. random saturation
5. random hue
6. convert color from HSV to BGR
7. random contrast (mode 1)
8. randomly swap channels

Parameters

- **brightness_delta** (*int*) – delta of brightness.
- **contrast_range** (*tuple*) – range of contrast.
- **saturation_range** (*tuple*) – range of saturation.
- **hue_delta** (*int*) – delta of hue.

brightness(*img*)

Brightness distortion.

contrast (*img*)

Contrast distortion.

convert (*img*, *alpha=1*, *beta=0*)

Multiple with alpha and add beta with clip.

class `mmpose.datasets.pipelines.shared_transform.RenameKeys` (*key_pairs*)

Rename the keys.

Parameters **key_pairs** (*Sequence[tuple]*) – Required keys to be renamed. If a tuple (*key_src*, *key_tgt*) is given as an element, the item retrieved by *key_src* will be renamed as *key_tgt*.

class `mmpose.datasets.pipelines.shared_transform.ToTensor` (*device='cpu'*)

Transform image to Tensor.

Required key: 'img'. Modifies key: 'img'.

Parameters **results** (*dict*) – contain all information about training.

class `mmpose.datasets.pipelines.top_down_transform.TopDownAffine` (*use_udp=False*)

Affine transform the image to make input.

Required key: 'img', 'joints_3d', 'joints_3d_visible', 'ann_info', 'scale', 'rotation' and 'center'.

Modified key: 'img', 'joints_3d', and 'joints_3d_visible'.

Parameters **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

class `mmpose.datasets.pipelines.top_down_transform.TopDownGenerateTarget` (*sigma=2*,
kernel=(11, 11),
valid_radius_factor=0.0546875,
target_type='GaussianHeatmap',
encoding='MSRA',
unbiased_encoding=False)

Generate the target heatmap.

Required key: 'joints_3d', 'joints_3d_visible', 'ann_info'.

Modified key: 'target', and 'target_weight'.

Parameters

- **sigma** – Sigma of heatmap gaussian for 'MSRA' approach.
- **kernel** – Kernel of heatmap gaussian for 'Megvii' approach.
- **encoding** (*str*) – Approach to generate target heatmaps. Currently supported approaches: 'MSRA', 'Megvii', 'UDP'. Default: 'MSRA'
- **unbiased_encoding** (*bool*) – Option to use unbiased encoding methods. Paper ref: Zhang et al. Distribution-Aware Coordinate Representation for Human Pose Estimation (CVPR 2020).
- **keypoint_pose_distance** – Keypoint pose distance for UDP. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).
- **target_type** (*str*) – supported targets: 'GaussianHeatmap', 'CombinedTarget'. Default: 'GaussianHeatmap' CombinedTarget: The combination of classification target (response map) and regression target (offset map). Paper ref: Huang et al. The Devil is in

the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

class `mmpose.datasets.pipelines.top_down_transform.TopDownGenerateTargetRegression`

Generate the target regression vector (coordinates).

Required key: 'joints_3d', 'joints_3d_visible', 'ann_info'. Modified key: 'target', and 'target_weight'.

class `mmpose.datasets.pipelines.top_down_transform.TopDownGetBboxCenterScale` (*padding: float = 1.25*)

Convert bbox from [x, y, w, h] to center and scale.

The center is the coordinates of the bbox center, and the scale is the bbox width and height normalized by a scale factor.

Required key: 'bbox', 'ann_info'

Modifies key: 'center', 'scale'

Parameters `padding` (*float*) – bbox padding scale that will be multilied to scale. Default: 1.25

class `mmpose.datasets.pipelines.top_down_transform.TopDownGetRandomScaleRotation` (*rot_factor=40, scale_factor=0.5, rot_prob=0.6*)

Data augmentation with random scaling & rotating.

Required key: 'scale'.

Modifies key: 'scale' and 'rotation'.

Parameters

- **rot_factor** (*int*) – Rotating to $[-2*\text{rot_factor}, 2*\text{rot_factor}]$.
- **scale_factor** (*float*) – Scaling to $[1-\text{scale_factor}, 1+\text{scale_factor}]$.
- **rot_prob** (*float*) – Probability of random rotation.

class `mmpose.datasets.pipelines.top_down_transform.TopDownHalfBodyTransform` (*num_joints_half_body=8, prob_half_body=0.3*)

Data augmentation with half-body transform. Keep only the upper body or the lower body at random.

Required key: 'joints_3d', 'joints_3d_visible', and 'ann_info'.

Modifies key: 'scale' and 'center'.

Parameters

- **num_joints_half_body** (*int*) – Threshold of performing half-body transform. If the body has fewer number of joints ($< \text{num_joints_half_body}$), ignore this step.
- **prob_half_body** (*float*) – Probability of half-body transform.

static `half_body_transform` (*cfg, joints_3d, joints_3d_visible*)

Get center&scale for half-body transform.

class `mmpose.datasets.pipelines.top_down_transform.TopDownRandomFlip` (*flip_prob=0.5*)

Data augmentation with random image flip.

Required key: 'img', 'joints_3d', 'joints_3d_visible', 'center' and 'ann_info'.

Modifies key: 'img', 'joints_3d', 'joints_3d_visible', 'center' and 'flipped'.

Parameters

- **flip** (*bool*) – Option to perform random flip.

- **flip_prob** (*float*) – Probability of flip.

```
class mmpose.datasets.pipelines.top_down_transform.TopDownRandomShiftBboxCenter(shift_factor:
                                                                    float = 0.16,
                                                                    prob: float
                                                                    = 0.3)
```

Random shift the bbox center.

Required key: 'center', 'scale'

Modifies key: 'center'

Parameters

- **shift_factor** (*float*) – The factor to control the shift range, which is $\text{scale} * \text{pixel_std} * \text{scale_factor}$. Default: 0.16
- **prob** (*float*) – Probability of applying random shift. Default: 0.3

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpGenerateHeatmapTarget(sigma,
                                                                    bg_weight=1.0,
                                                                    gen_center_heatmap=False,
                                                                    use_udp=False)
```

Generate multi-scale heatmap target for bottom-up.

Required key: 'joints', 'mask' and 'center'.

Modifies key: 'target', 'heatmaps' and 'masks'.

Parameters

- **sigma** (*int or tuple*) – Sigma of heatmap Gaussian. If sigma is a tuple, the first item should be the sigma of keypoints and the second item should be the sigma of center.
- **bg_weight** (*float*) – Weight for background. Default: 1.0.
- **gen_center_heatmap** (*bool*) – Whether to generate heatmaps for instance centers. Default: False.
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpGenerateOffsetTarget(radius=4)
Generate multi-scale offset target for bottom-up.
```

Required key: 'center', 'joints' and 'area'.

Modifies key: 'offsets', 'offset_weights'.

Parameters **radius** (*int*) – Radius of labeled area for each instance.

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpGeneratePAFTarget(limb_width,
                                                                    skele-
                                                                    ton=None)
```

Generate multi-scale heatmaps and part affinity fields (PAF) target for bottom-up. Paper ref: Cao et al. Realtime Multi-Person 2D Human Pose Estimation using Part Affinity Fields (CVPR 2017).

Parameters **limb_width** (*int*) – Limb width of part affinity fields

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpGenerateTarget(sigma,
                                                                    max_num_people,
                                                                    use_udp=False)
```

Generate multi-scale heatmap target for associate embedding.

Parameters

- **sigma** (*int*) – Sigma of heatmap Gaussian
- **max_num_people** (*int*) – Maximum number of people in an image
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpGetImgSize(test_scale_factor,  
                                                                    current_scale=1,  
                                                                    base_length=64,  
                                                                    use_udp=False)
```

Get multi-scale image sizes for bottom-up, including *base_size* and *test_scale_factor*. Keep the ratio and the image is resized to *results['ann_info']['image_size'] × current_scale*.

Parameters

- **test_scale_factor** (*List[float]*) – Multi scale
- **current_scale** (*int*) – default 1
- **base_length** (*int*) – The width and height should be multiples of *base_length*. Default: 64.
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpRandomAffine(rot_factor,  
                                                                    scale_factor,  
                                                                    scale_type,  
                                                                    trans_factor,  
                                                                    use_udp=False)
```

Data augmentation with random scaling & rotating.

Parameters

- **rot_factor** (*int*) – Rotating to [*-rotation_factor*, *rotation_factor*]
- **scale_factor** (*float*) – Scaling to [*1-scale_factor*, *1+scale_factor*]
- **scale_type** – wrt long or short length of the image.
- **trans_factor** – Translation factor.
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpRandomFlip(flip_prob=0.5)
```

Data augmentation with random image flip for bottom-up.

Parameters **flip_prob** (*float*) – Probability of flip.

```
class mmpose.datasets.pipelines.bottom_up_transform.BottomUpResizeAlign(transforms,  
                                                                    base_length=64,  
                                                                    use_udp=False)
```

Resize multi-scale size and align transform for bottom-up.

Parameters

- **transforms** (*List*) – ToTensor & Normalize

- **base_length** (*int*) – The width and height should be multiples of base_length. Default: 64.
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

class `mmpose.datasets.pipelines.bottom_up_transform.CIDGenerateTarget` (*max_num_people*)
Generate target for CID training.

Parameters **max_num_people** (*int*) – Maximum number of people in an image

class `mmpose.datasets.pipelines.bottom_up_transform.GetKeypointCenterArea` (*minimal_area=32*)
Compute center and area from keypoints for each instance.

Required key: 'joints'.

Modifies key: 'center' and 'area'.

Parameters **minimal_area** (*float*) – Minimum of allowed area. Instance with smaller area will be ignored in training. Default: 32.

class `mmpose.datasets.pipelines.bottom_up_transform.HeatmapGenerator` (*output_size, num_joints, sigma=-1, use_udp=False*)

Generate heatmaps for bottom-up models.

Parameters

- **num_joints** (*int*) – Number of keypoints
- **output_size** (*np.ndarray*) – Size (w, h) of feature map
- **sigma** (*int*) – Sigma of the heatmaps.
- **use_udp** (*bool*) – To use unbiased data processing. Paper ref: Huang et al. The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation (CVPR 2020).

class `mmpose.datasets.pipelines.bottom_up_transform.JointsEncoder` (*max_num_people, num_joints, output_size, tag_per_joint*)

Encodes the visible joints into (coordinates, score); The coordinate of one joint and its score are of *int* type.

(idx * output_size**2 + y * output_size + x, 1) or (0, 0).

Parameters

- **max_num_people** (*int*) – Max number of people in an image
- **num_joints** (*int*) – Number of keypoints
- **output_size** (*np.ndarray*) – Size (w, h) of feature map
- **tag_per_joint** (*bool*) – Option to use one tag map per joint.

class `mmpose.datasets.pipelines.bottom_up_transform.OffsetGenerator` (*output_size, num_joints, radius=4*)

Generate offset maps for bottom-up models.

Parameters

- **num_joints** (*int*) – Number of keypoints
- **output_size** (*np.ndarray*) – Size (w, h) of feature map
- **radius** (*int*) – Radius of area assigned with valid offset

class `mmpose.datasets.pipelines.bottom_up_transform.PAFGenerator`(*output_size, limb_width, skeleton*)

Generate part affinity fields.

Parameters

- **output_size** (*np.ndarray*) – Size (w, h) of feature map.
- **limb_width** (*int*) – Limb width of part affinity fields.
- **skeleton** (*list[list]*) – connections of joints.

class `mmpose.datasets.pipelines.mesh_transform.IUVToTensor`

Transform IUV image to part index mask and uv coordinates image. The 3 channels of IUV image means: part index, u coordinates, v coordinates.

Required key: 'iuv', 'ann_info'. Modifies key: 'part_index', 'uv_coordinates'.

Parameters **results** (*dict*) – contain all information about training.

class `mmpose.datasets.pipelines.mesh_transform.LoadIUVFromFile`(*to_float32=False*)

Loading IUV image from file.

class `mmpose.datasets.pipelines.mesh_transform.MeshAffine`

Affine transform the image to get input image. Affine transform the 2D keypoints, 3D keypoints and IUV image too.

Required keys: 'img', 'joints_2d', 'joints_2d_visible', 'joints_3d', 'joints_3d_visible', 'pose', 'iuv', 'ann_info', 'scale', 'rotation' and 'center'. Modifies key: 'img', 'joints_2d', 'joints_2d_visible', 'joints_3d', 'pose', 'iuv'.

class `mmpose.datasets.pipelines.mesh_transform.MeshGetRandomScaleRotation`(*rot_factor=30, scale_factor=0.25, rot_prob=0.6*)

Data augmentation with random scaling & rotating.

Required key: 'scale'. Modifies key: 'scale' and 'rotation'.

Parameters

- **rot_factor** (*int*) – Rotating to $[-2 * \text{rot_factor}, 2 * \text{rot_factor}]$.
- **scale_factor** (*float*) – Scaling to $[1 - \text{scale_factor}, 1 + \text{scale_factor}]$.
- **rot_prob** (*float*) – Probability of random rotation.

class `mmpose.datasets.pipelines.mesh_transform.MeshRandomChannelNoise`(*noise_factor=0.4*)

Data augmentation with random channel noise.

Required keys: 'img' Modifies key: 'img'

Parameters **noise_factor** (*float*) – Multiply each channel with a factor between $[1 - \text{scale_factor}, 1 + \text{scale_factor}]$

class `mmpose.datasets.pipelines.mesh_transform.MeshRandomFlip`(*flip_prob=0.5*)

Data augmentation with random image flip.

Required keys: 'img', 'joints_2d', 'joints_2d_visible', 'joints_3d', 'joints_3d_visible', 'center', 'pose', 'iuv' and 'ann_info'. Modifies key: 'img', 'joints_2d', 'joints_2d_visible', 'joints_3d', 'joints_3d_visible', 'center', 'pose', 'iuv'.

Parameters **flip_prob** (*float*) – Probability of flip.

class `mmpose.datasets.pipelines.pose3d_transform.AffineJoints`(*item='joints', visible_item=None*)

Apply affine transformation to joints coordinates.

Parameters

- **item** (*str*) – The name of the joints to apply affine.
- **visible_item** (*str*) – The name of the visibility item.

Required keys: item, visible_item(optional)

Modified keys: item, visible_item(optional)

```
class mmpose.datasets.pipelines.pose3d_transform.CameraProjection(item, mode,
                                                                output_name=None, cam-
                                                                era_type='SimpleCamera',
                                                                camera_param=None)
```

Apply camera projection to joint coordinates.

Parameters

- **item** (*str*) – The name of the pose to apply camera projection.
- **mode** (*str*) – The type of camera projection, supported options are
 - world_to_camera
 - world_to_pixel
 - camera_to_world
 - camera_to_pixel
- **output_name** (*str/None*) – The name of the projected pose. If None (default) is given, the projected pose will be stored in place.
- **camera_type** (*str*) – The camera class name (should be registered in CAMERA).
- **camera_param** (*dict/None*) – The camera parameter dict. See the camera class definition for more details. If None is given, the camera parameter will be obtained during processing of each data sample with the key “camera_param”.

Required keys:

- item
- camera_param (if camera parameters are not given in initialization)

Modified keys: output_name

```
class mmpose.datasets.pipelines.pose3d_transform.CollectCameraIntrinsics(camera_param=None,
                                                                need_distortion=True)
```

Store camera intrinsics in a 1-dim array, including f, c, k, p.

Parameters

- **camera_param** (*dict/None*) – The camera parameter dict. See the camera class definition for more details. If None is given, the camera parameter will be obtained during processing of each data sample with the key “camera_param”.
- **need_distortion** (*bool*) – Whether need distortion parameters k and p. Default: True.

Required keys: camera_param (if camera parameters are not given in initialization)

Modified keys: intrinsics

```
class mmpose.datasets.pipelines.pose3d_transform.Generate3DHeatmapTarget(sigma=2,  
                                                                    joint_indices=None,  
                                                                    max_bound=1.0)
```

Generate the target 3d heatmap.

Required keys: 'joints_3d', 'joints_3d_visible', 'ann_info'. Modified keys: 'target', and 'target_weight'.

Parameters

- **sigma** – Sigma of heatmap gaussian.
- **joint_indices** (*list*) – Indices of joints used for heatmap generation. If None (default) is given, all joints will be used.
- **max_bound** (*float*) – The maximal value of heatmap.

```
class mmpose.datasets.pipelines.pose3d_transform.GenerateInputHeatmaps(item='joints',  
                                                                    visible_item=None,  
                                                                    obscured=0.0,  
                                                                    from_pred=True,  
                                                                    sigma=3, scale=None,  
                                                                    base_size=96,  
                                                                    target_type='gaussian',  
                                                                    heatmap_cfg=None)
```

Generate 2D input heatmaps for multi-camera heatmaps when the 2D model is not available.

Required keys: 'joints' Modified keys: 'input_heatmaps'

Parameters

- **sigma** (*int*) – Sigma of heatmap gaussian (mm).
- **base_size** (*int*) – the base size of human
- **target_type** (*str*) – type of target heatmap, only support 'gaussian' now

```
class mmpose.datasets.pipelines.pose3d_transform.GenerateVoxel3DHeatmapTarget(sigma=200.0,  
                                                                    joint_indices=None)
```

Generate the target 3d heatmap.

Required keys: 'joints_3d', 'joints_3d_visible', 'ann_info_3d'. Modified keys: 'target', and 'target_weight'.

Parameters

- **sigma** – Sigma of heatmap gaussian (mm).
- **joint_indices** (*list*) – Indices of joints used for heatmap generation. If None (default) is given, all joints will be used.

```
class mmpose.datasets.pipelines.pose3d_transform.GetRootCenteredPose(item, root_index,  
                                                                    visible_item=None,  
                                                                    remove_root=False,  
                                                                    root_name=None)
```

Zero-center the pose around a given root joint. Optionally, the root joint can be removed from the original pose and stored as a separate item.

Note that the root-centered joints may no longer align with some annotation information (e.g. flip_pairs, num_joints, inference_channel, etc.) due to the removal of the root joint.

Parameters

- **item** (*str*) – The name of the pose to apply root-centering.
- **root_index** (*int*) – Root joint index in the pose.

- **visible_item** (*str*) – The name of the visibility item.
- **remove_root** (*bool*) – If true, remove the root joint from the pose
- **root_name** (*str*) – Optional. If not none, it will be used as the key to store the root position separated from the original pose.

Required keys: item

Modified keys: item, visible_item, root_name

```
class mmpose.datasets.pipelines.pose3d_transform.ImageCoordinateNormalization(item,
                                                                              norm_camera=False,
                                                                              cam-
                                                                              era_param=None)
```

Normalize the 2D joint coordinate with image width and height. Range [0, w] is mapped to [-1, 1], while preserving the aspect ratio.

Parameters

- **item** (*str/list[str]*) – The name of the pose to normalize.
- **norm_camera** (*bool*) – Whether to normalize camera intrinsics. Default: False.
- **camera_param** (*dict/None*) – The camera parameter dict. See the camera class definition for more details. If None is given, the camera parameter will be obtained during processing of each data sample with the key “camera_param”.

Required keys: item

Modified keys: item (, camera_param)

```
class mmpose.datasets.pipelines.pose3d_transform.NormalizeJointCoordinate(item, mean=None,
                                                                           std=None,
                                                                           norm_param_file=None)
```

Normalize the joint coordinate with given mean and std.

Parameters

- **item** (*str*) – The name of the pose to normalize.
- **mean** (*array*) – Mean values of joint coordinates in shape [K, C].
- **std** (*array*) – Std values of joint coordinates in shape [K, C].
- **norm_param_file** (*str*) – Optionally load a dict containing *mean* and *std* from a file using *mmcv.load*.

Required keys: item

Modified keys: item

```
class mmpose.datasets.pipelines.pose3d_transform.PoseSequenceToTensor(item)
```

Convert pose sequence from numpy array to Tensor.

The original pose sequence should have a shape of [T,K,C] or [K,C], where T is the sequence length, K and C are keypoint number and dimension. The converted pose sequence will have a shape of [KxC, T].

Parameters **item** (*str*) – The name of the pose sequence

Required keys: item

Modified keys: item

```
class mmpose.datasets.pipelines.pose3d_transform.RelativeJointRandomFlip(item, flip_cfg,  
                                                                    visible_item=None,  
                                                                    flip_prob=0.5,  
                                                                    flip_camera=False,  
                                                                    camera_param=None)
```

Data augmentation with random horizontal joint flip around a root joint.

Parameters

- **item** (*str/list[str]*) – The name of the pose to flip.
- **flip_cfg** (*dict/list[dict]*) – Configurations of the `flip_regression` function. It should contain the following arguments:
 - **center_mode**: The mode to set the center location on the x-axis to flip around.
 - **center_x** or **center_index**: Set the x-axis location or the root joint’s index to define the flip center.

Please refer to the docstring of the `flip_regression` function for more details.

- **visible_item** (*str/list[str]*) – The name of the visibility item which will be flipped accordingly along with the pose.
- **flip_prob** (*float*) – Probability of flip.
- **flip_camera** (*bool*) – Whether to flip horizontal distortion coefficients.
- **camera_param** (*dict/None*) – The camera parameter dict. See the camera class definition for more details. If `None` is given, the camera parameter will be obtained during processing of each data sample with the key “`camera_param`”.

Required keys: item

Modified keys: item (, camera_param)

50.3 samplers

```
class mmpose.datasets.samplers.DistributedSampler(dataset, num_replicas=None, rank=None,  
                                                shuffle=True, seed=0)
```

`DistributedSampler` inheriting from `torch.utils.data.DistributedSampler`.

In pytorch of lower versions, there is no `shuffle` argument. This child class will port one to `DistributedSampler`.

MMPOSE.UTILS

class `mmpose.utils.StopWatch(window=1)`

A helper class to measure FPS and detailed time consuming of each phase in a video processing loop or similar scenarios.

Parameters `window` (*int*) – The sliding window size to calculate the running average of the time consuming.

Example

```
>>> from mmpose.utils import StopWatch
>>> import time
>>> stop_watch = StopWatch(window=10)
>>> with stop_watch.timeit('total'):
>>>     time.sleep(0.1)
>>>     # 'timeit' support nested use
>>>     with stop_watch.timeit('phase1'):
>>>         time.sleep(0.1)
>>>     with stop_watch.timeit('phase2'):
>>>         time.sleep(0.2)
>>>     time.sleep(0.2)
>>> report = stop_watch.report()
```

report(*key=None*)

Report timing information.

Returns The key is the timer name and the value is the corresponding average time consuming.

Return type dict

report_strings()

Report timing information in texture strings.

Returns Each element is the information string of a timed event, in format of '{timer_name}:\n{time_in_ms}'. Specially, if timer_name is '_FPS_', the result will be converted to fps.

Return type list(str)

timeit(*timer_name='_FPS_'*)

Timing a code snippet with an assigned name.

Parameters `timer_name` (*str*) – The unique name of the interested code snippet to handle multiple timers and generate reports. Note that '_FPS_' is a special key that the measurement will be in *fps* instead of *millisecond*. Also see *report* and *report_strings*. Default: '_FPS_'.

Note: This function should always be used in a *with* statement, as shown in the example.

`mmpose.utils.get_root_logger(log_file=None, log_level=20)`

Use `get_logger` method in `mmdcv` to get the root logger.

The logger will be initialized if it has not been initialized. By default a `StreamHandler` will be added. If `log_file` is specified, a `FileHandler` will also be added. The name of the root logger is the top-level package name, e.g., “mmpose”.

Parameters

- **log_file** (*str* / *None*) – The log filename. If specified, a `FileHandler` will be added to the root logger.
- **log_level** (*int*) – The root logger level. Note that only the process of rank 0 is affected, while other processes will set the level to “Error” and be silent most of the time.

Returns The root logger.

Return type `logging.Logger`

`mmpose.utils.setup_multi_processes(cfg)`

Setup multi-processing environment variables.

CHAPTER
FIFTYTHREE

INDICES AND TABLES

- `genindex`
- `search`

PYTHON MODULE INDEX

m

- `mmpose.apis`, 591
- `mmpose.core.evaluation`, 627
- `mmpose.core.fp16`, 636
- `mmpose.core.post_processing`, 639
- `mmpose.core.utils`, 638
- `mmpose.datasets`, 739
- `mmpose.datasets.pipelines`, 808
 - `mmpose.datasets.pipelines.bottom_up_transform`, 813
 - `mmpose.datasets.pipelines.loading`, 808
 - `mmpose.datasets.pipelines.mesh_transform`, 816
 - `mmpose.datasets.pipelines.pose3d_transform`, 816
 - `mmpose.datasets.pipelines.shared_transform`, 808
 - `mmpose.datasets.pipelines.top_down_transform`, 811
- `mmpose.models.backbones`, 645
- `mmpose.models.detectors`, 679
- `mmpose.models.heads`, 707
- `mmpose.models.losses`, 726
- `mmpose.models.misc`, 738
- `mmpose.models.necks`, 675
- `mmpose.utils`, 821

Symbols

`__init__()` (*mmpose.apis.webcam.WebcamExecutor* method), 600
`__init__()` (*mmpose.apis.webcam.utils.BufferManager* method), 619
`__init__()` (*mmpose.apis.webcam.utils.EventManager* method), 622
`__init__()` (*mmpose.apis.webcam.utils.FrameMessage* method), 620
`__init__()` (*mmpose.apis.webcam.utils.Message* method), 619
`__init__()` (*mmpose.apis.webcam.utils.VideoEndingMessage* method), 620

A

AdaptiveWingLoss (class in *mmpose.models.losses*), 727
`adjust_width_group()` (*mmpose.models.backbones.RegNet* method), 658
AEHigherResolutionHead (class in *mmpose.models.heads*), 707
AELoss (class in *mmpose.models.losses*), 726
AEMultiStageHead (class in *mmpose.models.heads*), 708
AESimpleHead (class in *mmpose.models.heads*), 709
`affine_transform()` (in module *mmpose.core.post_processing*), 640
AffineJoints (class in *mmpose.datasets.pipelines.pose3d_transform*), 816
`after_train_iter()` (*mmpose.core.fp16.Fp16OptimizerHook* method), 636
`aggregate_scale()` (in module *mmpose.core.evaluation*), 627
`aggregate_stage_flip()` (in module *mmpose.core.evaluation*), 627
`albu_builder()` (*mmpose.datasets.pipelines.shared_transform.Albumentation* method), 809

Albumentation (class in *mmpose.datasets.pipelines.shared_transform*), 808
AlexNet (class in *mmpose.models.backbones*), 645
`allreduce_grads()` (in module *mmpose.core.utils*), 638
AnimalATRWDataset (class in *mmpose.datasets*), 739
AnimalFlyDataset (class in *mmpose.datasets*), 740
AnimalHorse10Dataset (class in *mmpose.datasets*), 742
AnimalLocustDataset (class in *mmpose.datasets*), 743
AnimalMacaqueDataset (class in *mmpose.datasets*), 745
AnimalPoseDataset (class in *mmpose.datasets*), 746
AnimalZebraDataset (class in *mmpose.datasets*), 747
`apply_bigeye_effect()` (*mmpose.apis.webcam.nodes.BigeyeEffectNode* method), 614
`apply_sunglasses_effect()` (*mmpose.apis.webcam.nodes.SunglassesEffectNode* method), 613
`assign2gt()` (*mmpose.models.detectors.VoxelCenterDetector* method), 703
AssociativeEmbedding (class in *mmpose.models.detectors*), 679
`auto_fp16()` (in module *mmpose.core.fp16*), 637

B

BaseVisualizerNode (class in *mmpose.apis.webcam.nodes*), 603
BCELoss (class in *mmpose.models.losses*), 728
`before_run()` (*mmpose.core.fp16.Fp16OptimizerHook* method), 636
BigeyeEffectNode (class in *mmpose.apis.webcam.nodes*), 614
Body3DH36MDataset (class in *mmpose.datasets*), 749
Body3DMviewDirectCampusDataset (class in *mmpose.datasets*), 750
Body3DMviewDirectPanopticDataset (class in *mmpose.datasets*), 751
Body3DMviewDirectShelfDataset (class in *mmpose.datasets*), 752

BoneLoss (class in *mmpose.models.losses*), 728
 BottomUpAicDataset (class in *mmpose.datasets*), 754
 BottomUpCocoDataset (class in *mmpose.datasets*), 754
 BottomUpCocoWholeBodyDataset (class in *mmpose.datasets*), 756
 BottomUpCrowdPoseDataset (class in *mmpose.datasets*), 756
 BottomUpGenerateHeatmapTarget (class in *mmpose.datasets.pipelines.bottom_up_transform*), 813
 BottomUpGenerateOffsetTarget (class in *mmpose.datasets.pipelines.bottom_up_transform*), 813
 BottomUpGeneratePAFTarget (class in *mmpose.datasets.pipelines.bottom_up_transform*), 813
 BottomUpGenerateTarget (class in *mmpose.datasets.pipelines.bottom_up_transform*), 813
 BottomUpGetImgSize (class in *mmpose.datasets.pipelines.bottom_up_transform*), 814
 BottomUpMhpDataset (class in *mmpose.datasets*), 757
 BottomUpRandomAffine (class in *mmpose.datasets.pipelines.bottom_up_transform*), 814
 BottomUpRandomFlip (class in *mmpose.datasets.pipelines.bottom_up_transform*), 814
 BottomUpResizeAlign (class in *mmpose.datasets.pipelines.bottom_up_transform*), 814
 brightness() (*mmpose.datasets.pipelines.shared_transform.PhotometricDistortion* method), 810
 BufferManager (class in *mmpose.apis.webcam.utils*), 619
 build_dataloader() (in module *mmpose.datasets*), 789
 build_dataset() (in module *mmpose.datasets*), 789
 build_sample_indices() (*mmpose.datasets.Body3DH36MDataset* method), 749
 bypass() (*mmpose.apis.webcam.nodes.BaseVisualizerNode* method), 603
 bypass() (*mmpose.apis.webcam.nodes.DetectorNode* method), 605
 bypass() (*mmpose.apis.webcam.nodes.MonitorNode* method), 617
 bypass() (*mmpose.apis.webcam.nodes.Node* method), 601
 bypass() (*mmpose.apis.webcam.nodes.PoseTrackerNode* method), 609
 bypass() (*mmpose.apis.webcam.nodes.TopDownPoseEstimatorNode* method), 607

C

CameraProjection (class in *mmpose.datasets.pipelines.pose3d_transform*), 817
 cast_tensor_type() (in module *mmpose.core.fp16*), 637
 CID (class in *mmpose.models.detectors*), 681
 CIDGenerateTarget (class in *mmpose.datasets.pipelines.bottom_up_transform*), 815
 CIDHead (class in *mmpose.models.heads*), 710
 coco2campus3D() (*mmpose.datasets.Body3DMviewDirectCampusDataset* static method), 750
 coco2shelf3D() (*mmpose.datasets.Body3DMviewDirectShelfDataset* static method), 753
 Collect (class in *mmpose.datasets.pipelines.shared_transform*), 809
 collect_multi_frames() (in module *mmpose.apis*), 591
 CollectCameraIntrinsics (class in *mmpose.datasets.pipelines.pose3d_transform*), 817
 Compose (class in *mmpose.datasets*), 757
 Compose (class in *mmpose.datasets.pipelines.shared_transform*), 809
 compute_similarity_transform() (in module *mmpose.core.evaluation*), 628
 contrast() (*mmpose.datasets.pipelines.shared_transform.PhotometricDistortion* method), 810
 convert() (*mmpose.datasets.pipelines.shared_transform.PhotometricDistortion* method), 811
 copy_and_paste() (in module *mmpose.apis.webcam.utils*), 623
 copy_grads_to_fp32() (*mmpose.core.fp16.Fp16OptimizerHook* static method), 637
 copy_params_to_fp16() (*mmpose.core.fp16.Fp16OptimizerHook* static method), 637
 CPM (class in *mmpose.models.backbones*), 645
 criterion() (*mmpose.models.losses.AdaptiveWingLoss* method), 727
 criterion() (*mmpose.models.losses.SoftWingLoss* method), 736
 criterion() (*mmpose.models.losses.WingLoss* method), 737
 CuboidCenterHead (class in *mmpose.models.heads*), 710
 CuboidPoseHead (class in *mmpose.models.heads*), 710

D

`decode()` (*mmpose.models.heads.DeepPoseRegressionHead* method), 713
`decode()` (*mmpose.models.heads.Interhand3DHead* method), 715
`decode()` (*mmpose.models.heads.TemporalRegressionHead* method), 718
`decode()` (*mmpose.models.heads.TopdownHeatmapBaseHead* method), 719
`DeconvHead` (class in *mmpose.models.heads*), 712
`DeepFashionDataset` (class in *mmpose.datasets*), 758
`DeepPoseRegressionHead` (class in *mmpose.models.heads*), 713
`DEKRHead` (class in *mmpose.models.heads*), 711
`DetectAndRegress` (class in *mmpose.models.detectors*), 685
`DetectorNode` (class in *mmpose.apis.webcam.nodes*), 604
`DisentangledKeypointRegressor` (class in *mmpose.models.detectors*), 687
`DistEvalHook` (class in *mmpose.core.evaluation*), 627
`DistributedSampler` (class in *mmpose.datasets*), 759
`draw()` (*mmpose.apis.webcam.nodes.BaseVisualizerNode* method), 603
`draw()` (*mmpose.apis.webcam.nodes.BigeyeEffectNode* method), 614
`draw()` (*mmpose.apis.webcam.nodes.NoticeBoardNode* method), 612
`draw()` (*mmpose.apis.webcam.nodes.ObjectVisualizerNode* method), 611
`draw()` (*mmpose.apis.webcam.nodes.SunglassesEffectNode* method), 613
`evaluate()` (*mmpose.datasets.Body3DMviewDirectPanopticDataset* method), 752
`evaluate()` (*mmpose.datasets.Body3DMviewDirectShelfDataset* method), 753
`evaluate()` (*mmpose.datasets.BottomUpCocoDataset* method), 755
`evaluate()` (*mmpose.datasets.DeepFashionDataset* method), 758
`evaluate()` (*mmpose.datasets.Face300WDataset* method), 760
`evaluate()` (*mmpose.datasets.FaceAFLWDataset* method), 761
`evaluate()` (*mmpose.datasets.FaceCocoWholeBodyDataset* method), 762
`evaluate()` (*mmpose.datasets.FaceCOFWDataset* method), 761
`evaluate()` (*mmpose.datasets.FaceWFLWDataset* method), 763
`evaluate()` (*mmpose.datasets.FreiHandDataset* method), 765
`evaluate()` (*mmpose.datasets.HandCocoWholeBodyDataset* method), 766
`evaluate()` (*mmpose.datasets.InterHand2DDataset* method), 768
`evaluate()` (*mmpose.datasets.InterHand3DDataset* method), 770
`evaluate()` (*mmpose.datasets.MeshH36MDataset* method), 771
`evaluate()` (*mmpose.datasets.NVGestureDataset* method), 772
`evaluate()` (*mmpose.datasets.OneHand10KDataset* method), 773
`evaluate()` (*mmpose.datasets.PanopticDataset* method), 775
`evaluate()` (*mmpose.datasets.TopDownCocoDataset* method), 777
`evaluate()` (*mmpose.datasets.TopDownFreiHandDataset* method), 779
`evaluate()` (*mmpose.datasets.TopDownH36MDataset* method), 779
`evaluate()` (*mmpose.datasets.TopDownJhmdbDataset* method), 781
`evaluate()` (*mmpose.datasets.TopDownMpiiDataset* method), 783
`evaluate()` (*mmpose.datasets.TopDownMpiiTrbDataset* method), 785
`evaluate()` (*mmpose.datasets.TopDownOneHand10KDataset* method), 786
`evaluate()` (*mmpose.datasets.TopDownPanopticDataset* method), 786
`evaluate()` (*mmpose.datasets.TopDownPoseTrack18Dataset* method), 787
`evaluate()` (*mmpose.datasets.TopDownPoseTrack18VideoDataset* method), 788

E

`EvalHook` (class in *mmpose.core.evaluation*), 627
`evaluate()` (*mmpose.datasets.AnimalATRWDataset* method), 739
`evaluate()` (*mmpose.datasets.AnimalFlyDataset* method), 741
`evaluate()` (*mmpose.datasets.AnimalHorse10Dataset* method), 742
`evaluate()` (*mmpose.datasets.AnimalLocustDataset* method), 744
`evaluate()` (*mmpose.datasets.AnimalMacaqueDataset* method), 746
`evaluate()` (*mmpose.datasets.AnimalPoseDataset* method), 747
`evaluate()` (*mmpose.datasets.AnimalZebraDataset* method), 748
`evaluate()` (*mmpose.datasets.Body3DH36MDataset* method), 749
`evaluate()` (*mmpose.datasets.Body3DMviewDirectCampusDataset* method), 750

EventManager (class in *mmpose.apis.webcam.utils*), 622
 expand_and_clamp() (in module *mmpose.apis.webcam.utils*), 624
 extract_pose_sequence() (in module *mmpose.apis*), 591

F

Face300WDataset (class in *mmpose.datasets*), 759
 FaceAFLWDataset (class in *mmpose.datasets*), 760
 FaceCocoWholeBodyDataset (class in *mmpose.datasets*), 762
 FaceCOFWDataset (class in *mmpose.datasets*), 761
 FaceWFLWDataset (class in *mmpose.datasets*), 763
 flip_back() (in module *mmpose.core.post_processing*), 640
 flip_feature_maps() (in module *mmpose.core.evaluation*), 628
 flipplr_joints() (in module *mmpose.core.post_processing*), 641
 flipplr_regression() (in module *mmpose.core.post_processing*), 641
 FocalHeatmapLoss (class in *mmpose.models.losses*), 728
 force_fp32() (in module *mmpose.core.fp16*), 637
 forward() (*mmpose.models.backbones.AlexNet* method), 645
 forward() (*mmpose.models.backbones.CPM* method), 646
 forward() (*mmpose.models.backbones.HourglassAENet* method), 650
 forward() (*mmpose.models.backbones.HourglassNet* method), 650
 forward() (*mmpose.models.backbones.HRNet* method), 649
 forward() (*mmpose.models.backbones.I3D* method), 651
 forward() (*mmpose.models.backbones.LiteHRNet* method), 652
 forward() (*mmpose.models.backbones.MobileNetV2* method), 653
 forward() (*mmpose.models.backbones.MobileNetV3* method), 654
 forward() (*mmpose.models.backbones.MSPN* method), 653
 forward() (*mmpose.models.backbones.PyramidVisionTransformer* method), 656
 forward() (*mmpose.models.backbones.RegNet* method), 658
 forward() (*mmpose.models.backbones.ResNet* method), 662
 forward() (*mmpose.models.backbones.RSN* method), 657
 forward() (*mmpose.models.backbones.ShuffleNetV1* method), 666

forward() (*mmpose.models.backbones.ShuffleNetV2* method), 667
 forward() (*mmpose.models.backbones.SwinTransformer* method), 669
 forward() (*mmpose.models.backbones.TCFormer* method), 670
 forward() (*mmpose.models.backbones.TCN* method), 671
 forward() (*mmpose.models.backbones.V2VNet* method), 671
 forward() (*mmpose.models.backbones.VGG* method), 672
 forward() (*mmpose.models.backbones.ViPNAS_MobileNetV3* method), 673
 forward() (*mmpose.models.backbones.ViPNAS_ResNet* method), 674
 forward() (*mmpose.models.detectors.AssociativeEmbedding* method), 679
 forward() (*mmpose.models.detectors.CID* method), 682
 forward() (*mmpose.models.detectors.DetectAndRegress* method), 685
 forward() (*mmpose.models.detectors.DisentangledKeypointRegressor* method), 687
 forward() (*mmpose.models.detectors.GestureRecognizer* method), 690
 forward() (*mmpose.models.detectors.Interhand3D* method), 691
 forward() (*mmpose.models.detectors.MultiTask* method), 693
 forward() (*mmpose.models.detectors.ParametricMesh* method), 695
 forward() (*mmpose.models.detectors.PoseLifter* method), 697
 forward() (*mmpose.models.detectors.PoseWarper* method), 700
 forward() (*mmpose.models.detectors.TopDown* method), 701
 forward() (*mmpose.models.detectors.VoxelCenterDetector* method), 703
 forward() (*mmpose.models.detectors.VoxelSinglePose* method), 705
 forward() (*mmpose.models.heads.AEHigherResolutionHead* method), 707
 forward() (*mmpose.models.heads.AEMultiStageHead* method), 708
 forward() (*mmpose.models.heads.CIDHead* method), 710
 forward() (*mmpose.models.heads.CuboidCenterHead* method), 710
 forward() (*mmpose.models.heads.CuboidPoseHead* method), 711
 forward() (*mmpose.models.heads.DeconvHead* method), 712
 forward() (*mmpose.models.heads.DeepPoseRegressionHead* method), 712

`method`), 713
`forward()` (`mmpose.models.heads.DEKRHead` `method`), 711
`forward()` (`mmpose.models.heads.HMRMeshHead` `method`), 714
`forward()` (`mmpose.models.heads.Interhand3DHead` `method`), 715
`forward()` (`mmpose.models.heads.MultiModalSSAHead` `method`), 716
`forward()` (`mmpose.models.heads.TemporalRegressionHead` `method`), 718
`forward()` (`mmpose.models.heads.TopdownHeatmapBaseHead` `method`), 719
`forward()` (`mmpose.models.heads.TopdownHeatmapMSMUHead` `method`), 720
`forward()` (`mmpose.models.heads.TopdownHeatmapMultiStageHead` `method`), 722
`forward()` (`mmpose.models.heads.TopdownHeatmapSimpleHead` `method`), 723
`forward()` (`mmpose.models.heads.ViPNASHeatmapSimpleHead` `method`), 725
`forward()` (`mmpose.models.losses.AdaptiveWingLoss` `method`), 727
`forward()` (`mmpose.models.losses.AELoss` `method`), 726
`forward()` (`mmpose.models.losses.BCELoss` `method`), 728
`forward()` (`mmpose.models.losses.BoneLoss` `method`), 728
`forward()` (`mmpose.models.losses.FocalHeatmapLoss` `method`), 728
`forward()` (`mmpose.models.losses.GANLoss` `method`), 729
`forward()` (`mmpose.models.losses.HeatmapLoss` `method`), 729
`forward()` (`mmpose.models.losses.JointsMSELoss` `method`), 730
`forward()` (`mmpose.models.losses.JointsOHKMMSELoss` `method`), 730
`forward()` (`mmpose.models.losses.L1Loss` `method`), 730
`forward()` (`mmpose.models.losses.MeshLoss` `method`), 732
`forward()` (`mmpose.models.losses.MPJPELoss` `method`), 731
`forward()` (`mmpose.models.losses.MSELoss` `method`), 731
`forward()` (`mmpose.models.losses.MultiLossFactory` `method`), 733
`forward()` (`mmpose.models.losses.RLELoss` `method`), 734
`forward()` (`mmpose.models.losses.SemiSupervisionLoss` `method`), 734
`forward()` (`mmpose.models.losses.SmoothL1Loss` `method`), 735
`forward()` (`mmpose.models.losses.SoftWeightSmoothL1Loss` `method`), 735
`forward()` (`mmpose.models.losses.SoftWingLoss` `method`), 736
`forward()` (`mmpose.models.losses.WingLoss` `method`), 737
`forward()` (`mmpose.models.necks.FPN` `method`), 676
`forward()` (`mmpose.models.necks.GlobalAveragePooling` `method`), 676
`forward()` (`mmpose.models.necks.MTA` `method`), 677
`forward()` (`mmpose.models.necks.PoseWarperNeck` `method`), 678
`forward_dummy()` (`mmpose.models.detectors.AssociativeEmbedding` `method`), 680
`forward_dummy()` (`mmpose.models.detectors.CID` `method`), 683
`forward_dummy()` (`mmpose.models.detectors.DetectAndRegress` `method`), 686
`forward_dummy()` (`mmpose.models.detectors.DisentangledKeypointRegressor` `method`), 688
`forward_dummy()` (`mmpose.models.detectors.MultiTask` `method`), 694
`forward_dummy()` (`mmpose.models.detectors.ParametricMesh` `method`), 695
`forward_dummy()` (`mmpose.models.detectors.PoseLifter` `method`), 698
`forward_dummy()` (`mmpose.models.detectors.PoseWarper` `method`), 701
`forward_dummy()` (`mmpose.models.detectors.TopDown` `method`), 702
`forward_dummy()` (`mmpose.models.detectors.VoxelCenterDetector` `method`), 704
`forward_dummy()` (`mmpose.models.detectors.VoxelSinglePose` `method`), 706
`forward_test()` (`mmpose.models.detectors.AssociativeEmbedding` `method`), 680
`forward_test()` (`mmpose.models.detectors.CID` `method`), 683
`forward_test()` (`mmpose.models.detectors.DetectAndRegress` `method`), 686
`forward_test()` (`mmpose.models.detectors.DisentangledKeypointRegressor` `method`), 688
`forward_test()` (`mmpose.models.detectors.GestureRecognizer` `method`), 691

`forward_test()` (*mm-pose.models.detectors.Interhand3D method*), 692
`forward_test()` (*mm-pose.models.detectors.MultiTask method*), 694
`forward_test()` (*mm-pose.models.detectors.ParametricMesh method*), 696
`forward_test()` (*mm-pose.models.detectors.PoseLifter method*), 698
`forward_test()` (*mm-pose.models.detectors.PoseWarper method*), 701
`forward_test()` (*mm-pose.models.detectors.TopDown method*), 702
`forward_test()` (*mm-pose.models.detectors.VoxelCenterDetector method*), 704
`forward_test()` (*mm-pose.models.detectors.VoxelSinglePose method*), 706
`forward_train()` (*mm-pose.models.detectors.AssociativeEmbedding method*), 680
`forward_train()` (*mm-pose.models.detectors.CID method*), 683
`forward_train()` (*mm-pose.models.detectors.DetectAndRegress method*), 686
`forward_train()` (*mm-pose.models.detectors.DisentangledKeypointRegressor method*), 689
`forward_train()` (*mm-pose.models.detectors.GestureRecognizer method*), 691
`forward_train()` (*mm-pose.models.detectors.MultiTask method*), 694
`forward_train()` (*mm-pose.models.detectors.ParametricMesh method*), 696
`forward_train()` (*mm-pose.models.detectors.PoseLifter method*), 698
`forward_train()` (*mm-pose.models.detectors.PoseWarper method*), 701
`forward_train()` (*mm-pose.models.detectors.TopDown method*), 702
`forward_train()` (*mm-pose.models.detectors.VoxelCenterDetector method*), 704
`forward_train()` (*mm-pose.models.detectors.VoxelSinglePose method*), 706
`Fp16OptimizerHook` (*class in mm-pose.core.fp16*), 636
`FPN` (*class in mm-pose.models.necks*), 675
`FrameMessage` (*class in mm-pose.apis.webcam.utils*), 620
`FreiHandDataset` (*class in mm-pose.datasets*), 764

G

`GANLoss` (*class in mm-pose.models.losses*), 729
`Generate3DHeatmapTarget` (*class in mm-pose.datasets.pipelines.pose3d_transform*), 817
`generate_regnet()` (*mm-pose.models.backbones.RegNet static method*), 658
`GenerateInputHeatmaps` (*class in mm-pose.datasets.pipelines.pose3d_transform*), 818
`GenerateVoxel3DHeatmapTarget` (*class in mm-pose.datasets.pipelines.pose3d_transform*), 818
`GestureRecognizer` (*class in mm-pose.models.detectors*), 690
`get_3d_joints_from_mesh()` (*mm-pose.models.detectors.ParametricMesh method*), 696
`get_accuracy()` (*mm-pose.models.heads.DeepPoseRegressionHead method*), 714
`get_accuracy()` (*mm-pose.models.heads.Interhand3DHead method*), 715
`get_accuracy()` (*mm-pose.models.heads.MultiModalSSAHead method*), 716
`get_accuracy()` (*mm-pose.models.heads.TemporalRegressionHead method*), 718
`get_accuracy()` (*mm-pose.models.heads.TopdownHeatmapBaseHead method*), 720
`get_accuracy()` (*mm-pose.models.heads.TopdownHeatmapMSMUHead method*), 720
`get_accuracy()` (*mm-pose.models.heads.TopdownHeatmapMultiStageHead method*), 722
`get_accuracy()` (*mm-pose.models.heads.TopdownHeatmapSimpleHead method*), 723
`get_accuracy()` (*mm-pose.models.heads.ViPNASHeatmapSimpleHead method*), 725
`get_affine_transform()` (*in module mm-pose.core.post_processing*), 642
`get_cached_file_path()` (*in module mm-pose.apis.webcam.utils*), 625

get_camera_param() (mm-pose.models.losses.GANLoss method), 729
 get_track_id() (in module mmpose.apis), 591
 get_warp_matrix() (in module mm-pose.core.post_processing), 642
 get_wrist_keypoint_ids() (in module mm-pose.apis.webcam.utils), 622
 GetKeypointCenterArea (class in mm-pose.datasets.pipelines.bottom_up_transform), 815
 GetRootCenteredPose (class in mm-pose.datasets.pipelines.pose3d_transform), 818
 GlobalAveragePooling (class in mm-pose.models.necks), 676
H
 half_body_transform() (mm-pose.datasets.pipelines.top_down_transform.TopDownHalfBodyTransform static method), 812
 HandCocoWholeBodyDataset (class in mm-pose.datasets), 765
 HeatmapGenerator (class in mm-pose.datasets.pipelines.bottom_up_transform), 815
 HeatmapLoss (class in mmpose.models.losses), 729
 HMRMeshHead (class in mmpose.models.heads), 714
 hook() (mmpose.core.utils.WeightNormClipHook method), 638
 hook_type (mmpose.core.utils.WeightNormClipHook property), 638
 HourglassAENet (class in mmpose.models.backbones), 649
 HourglassNet (class in mmpose.models.backbones), 650
 HRFormer (class in mmpose.models.backbones), 646
 HRNet (class in mmpose.models.backbones), 648
I
 ImageCoordinateNormalization (class in mm-pose.datasets.pipelines.pose3d_transform), 819
 inference_bottom_up_pose_model() (in module mmpose.apis), 592
 inference_interhand_3d_model() (in module mmpose.apis), 593
 inference_mesh_model() (in module mmpose.apis), 593
 inference_model() (mmpose.models.heads.DeepPoseRegressionHead method), 714
 inference_model() (mmpose.models.heads.Interhand3DHead method), 716
 get_config_path() (in module mm-pose.apis.webcam.utils), 626
 get_eye_keypoint_ids() (in module mm-pose.apis.webcam.utils), 621
 get_face_keypoint_ids() (in module mm-pose.apis.webcam.utils), 621
 get_group_preds() (in module mm-pose.core.evaluation), 629
 get_hand_keypoint_ids() (in module mm-pose.apis.webcam.utils), 622
 get_loss() (mmpose.models.heads.AEHigherResolutionHead method), 707
 get_loss() (mmpose.models.heads.AEMultiStageHead method), 708
 get_loss() (mmpose.models.heads.AESimpleHead method), 709
 get_loss() (mmpose.models.heads.DeconvHead method), 712
 get_loss() (mmpose.models.heads.DeepPoseRegressionHead method), 714
 get_loss() (mmpose.models.heads.DEKRHead method), 711
 get_loss() (mmpose.models.heads.Interhand3DHead method), 716
 get_loss() (mmpose.models.heads.MultiModalSSAHead method), 717
 get_loss() (mmpose.models.heads.TemporalRegressionHead method), 718
 get_loss() (mmpose.models.heads.TopdownHeatmapBaseHead method), 720
 get_loss() (mmpose.models.heads.TopdownHeatmapMSMHHead method), 721
 get_loss() (mmpose.models.heads.TopdownHeatmapMultiStageHead method), 722
 get_loss() (mmpose.models.heads.TopdownHeatmapSimpleHead method), 724
 get_loss() (mmpose.models.heads.ViPNASHHeatmapSimpleHead method), 725
 get_mouth_keypoint_ids() (in module mm-pose.apis.webcam.utils), 622
 get_new_center() (mm-pose.datasets.Body3DViewDirectCampusDataset static method), 751
 get_new_center() (mm-pose.datasets.Body3DViewDirectShelfDataset static method), 753
 get_root_logger() (in module mmpose.utils), 822
 get_stages_from_blocks() (mmpose.models.backbones.RegNet method), 659
 get_target_label() (mm-

<code>inference_model()</code>	(<code>mm-pose.models.heads.TemporalRegressionHead</code> method), 719	<code>init_weights()</code>	(<code>mm-pose.models.backbones.ShuffleNetV2</code> method), 666
<code>inference_model()</code>	(<code>mm-pose.models.heads.TopdownHeatmapBaseHead</code> method), 720	<code>init_weights()</code>	(<code>mm-pose.models.backbones.SwinTransformer</code> method), 667
<code>inference_model()</code>	(<code>mm-pose.models.heads.TopdownHeatmapMSMUHead</code> method), 721	<code>init_weights()</code>	(<code>mmpose.models.backbones.TCN</code> method), 669
<code>inference_model()</code>	(<code>mm-pose.models.heads.TopdownHeatmapMultiStageHead</code> method), 722	<code>init_weights()</code>	(<code>mmpose.models.backbones.VGG</code> method), 671
<code>inference_model()</code>	(<code>mm-pose.models.heads.TopdownHeatmapSimpleHead</code> method), 724	<code>init_weights()</code>	(<code>mm-pose.models.backbones.ViPNAS_MobileNetV3</code> method), 672
<code>inference_model()</code>	(<code>mm-pose.models.heads.ViPNASHeatmapSimpleHead</code> method), 726	<code>init_weights()</code>	(<code>mm-pose.models.backbones.ViPNAS_ResNet</code> method), 673
<code>inference_pose_lifter_model()</code> (in module <code>mm-pose.apis</code>), 594		<code>init_weights()</code>	(<code>mm-pose.models.detectors.AssociativeEmbedding</code> method), 674
<code>inference_top_down_pose_model()</code> (in module <code>mm-pose.apis</code>), 595		<code>init_weights()</code>	(<code>mmpose.models.detectors.CID</code> method), 681
<code>init_pose_model()</code> (in module <code>mmpose.apis</code>), 596		<code>init_weights()</code>	(<code>mm-pose.models.detectors.DisentangledKeypointRegressor</code> method), 684
<code>init_random_seed()</code> (in module <code>mmpose.apis</code>), 596		<code>init_weights()</code>	(<code>mm-pose.models.detectors.GestureRecognizer</code> method), 689
<code>init_weights()</code> (<code>mmpose.models.backbones.CPM</code> method), 646		<code>init_weights()</code>	(<code>mm-pose.models.detectors.MultiTask</code> method), 691
<code>init_weights()</code> (<code>mm-pose.models.backbones.HourglassAENet</code> method), 650		<code>init_weights()</code>	(<code>mm-pose.models.detectors.ParametricMesh</code> method), 694
<code>init_weights()</code> (<code>mm-pose.models.backbones.HourglassNet</code> method), 651		<code>init_weights()</code>	(<code>mm-pose.models.detectors.PoseLifter</code> method), 696
<code>init_weights()</code> (<code>mmpose.models.backbones.HRNet</code> method), 649		<code>init_weights()</code>	(<code>mmpose.models.detectors.TopDown</code> method), 698
<code>init_weights()</code> (<code>mm-pose.models.backbones.LiteHRNet</code> method), 652		<code>init_weights()</code>	(<code>mm-pose.models.heads.AEHigherResolutionHead</code> method), 702
<code>init_weights()</code> (<code>mm-pose.models.backbones.MobileNetV2</code> method), 653		<code>init_weights()</code>	(<code>mm-pose.models.heads.AEMultiStageHead</code> method), 708
<code>init_weights()</code> (<code>mm-pose.models.backbones.MobileNetV3</code> method), 654		<code>init_weights()</code>	(<code>mmpose.models.heads.CIDHead</code> method), 709
<code>init_weights()</code> (<code>mmpose.models.backbones.MSPN</code> method), 653		<code>init_weights()</code>	(<code>mmpose.models.heads.DeconvHead</code> method), 710
<code>init_weights()</code> (<code>mm-pose.models.backbones.PyramidVisionTransformer</code> method), 656		<code>init_weights()</code>	(<code>mmpose.models.heads.DEKRHead</code> method), 713
<code>init_weights()</code> (<code>mmpose.models.backbones.ResNet</code> method), 662		<code>init_weights()</code>	(<code>mm-pose.models.heads.HMRMeshHead</code> method), 712
<code>init_weights()</code> (<code>mmpose.models.backbones.RSN</code> method), 657		<code>init_weights()</code>	(<code>mm-pose.models.heads.MultiModalSSAHead</code> method), 715
<code>init_weights()</code> (<code>mm-pose.models.backbones.ShuffleNetV1</code> method),		<code>init_weights()</code>	

method), 717
 init_weights() (mm-pose.models.heads.TemporalRegressionHead method), 719
 init_weights() (mm-pose.models.heads.TopdownHeatmapMSMUHead method), 721
 init_weights() (mm-pose.models.heads.TopdownHeatmapMultiStageHead method), 723
 init_weights() (mm-pose.models.heads.TopdownHeatmapSimpleHead method), 724
 init_weights() (mm-pose.models.heads.ViPNASHeatmapSimpleHead method), 726
 init_weights() (mmpose.models.necks.FPN method), 676
 init_weights() (mmpose.models.necks.MTA method), 677
 InterHand2DDataset (class in mmpose.datasets), 767
 Interhand3D (class in mmpose.models.detectors), 691
 InterHand3DDataset (class in mmpose.datasets), 768
 Interhand3DHead (class in mmpose.models.heads), 715
 is_image_file() (in module mm-pose.apis.webcam.utils), 625
 isvalid() (mmpose.datasets.Body3DMviewDirectCampusDataset method), 751
 isvalid() (mmpose.datasets.Body3DMviewDirectShelfDataset static method), 753
 IUVTensor (class in mmpose.datasets.pipelines.mesh_transform), 816
J
 joints_2d_loss() (mmpose.models.losses.MeshLoss method), 732
 joints_3d_loss() (mmpose.models.losses.MeshLoss method), 732
 JointsEncoder (class in mmpose.datasets.pipelines.bottom_up_transform), 815
 JointsMSELoss (class in mmpose.models.losses), 730
 JointsOHKMMSELoss (class in mmpose.models.losses), 730
K
 keypoint_3d_auc() (in module mm-pose.core.evaluation), 629
 keypoint_3d_pck() (in module mm-pose.core.evaluation), 629
 keypoint_auc() (in module mmpose.core.evaluation), 630
 keypoint_epe() (in module mmpose.core.evaluation), 630
 keypoint_mpjpe() (in module mm-pose.core.evaluation), 631
 keypoint_pck_accuracy() (in module mm-pose.core.evaluation), 631
 keypoints_from_heatmaps() (in module mm-pose.core.evaluation), 632
 keypoints_from_heatmaps3d() (in module mm-pose.core.evaluation), 633
 keypoints_from_regression() (in module mm-pose.core.evaluation), 633
L
 L1Loss (class in mmpose.models.losses), 730
 limit_max_fps() (in module mm-pose.apis.webcam.utils), 624
 LiteHRNet (class in mmpose.models.backbones), 651
 load_annotations() (mmpose.datasets.Body3DH36MDataset method), 749
 load_config() (mmpose.datasets.Body3DH36MDataset method), 749
 load_config() (mmpose.datasets.Body3DMviewDirectCampusDataset method), 751
 load_config() (mmpose.datasets.Body3DMviewDirectPanopticDataset method), 752
 load_config() (mmpose.datasets.Body3DMviewDirectShelfDataset method), 754
 load_image_from_disk_or_url() (in module mm-pose.apis.webcam.utils), 625
 LoadImageFromFile (class in mmpose.datasets.pipelines.loading), 808
 LoadIUVFromFile (class in mmpose.datasets.pipelines.mesh_transform), 816
 LoadVideoFromFile (class in mmpose.datasets.pipelines.loading), 808
M
 make_layer() (mmpose.models.backbones.MobileNetV2 method), 654
 make_layer() (mmpose.models.backbones.ShuffleNetV1 method), 666
 make_res_layer() (mmpose.models.backbones.ResNeSt method), 660
 make_res_layer() (mmpose.models.backbones.ResNet method), 662
 make_res_layer() (mmpose.models.backbones.ResNeXt method), 661
 make_res_layer() (mmpose.models.backbones.SEResNet method),

666
 make_res_layer() (mmpose.models.backbones.SEResNeXt method), 664
 make_res_layer() (mmpose.models.backbones.ViPNAS_ResNet method), 674
 mapper() (mmpose.datasets.pipelines.shared_transform.AllImageToImage module), 809
 MeshAdversarialDataset (class in mmpose.datasets), 770
 MeshAffine (class in mmpose.datasets.pipelines.mesh_transform), 816
 MeshGetRandomScaleRotation (class in mmpose.datasets.pipelines.mesh_transform), 816
 MeshH36MDataset (class in mmpose.datasets), 770
 MeshLoss (class in mmpose.models.losses), 731
 MeshMixDataset (class in mmpose.datasets), 771
 MeshRandomChannelNoise (class in mmpose.datasets.pipelines.mesh_transform), 816
 MeshRandomFlip (class in mmpose.datasets.pipelines.mesh_transform), 816
 Message (class in mmpose.apis.webcam.utils), 619
 mmpose.apis module, 591
 mmpose.core.evaluation module, 627
 mmpose.core.fp16 module, 636
 mmpose.core.post_processing module, 639
 mmpose.core.utils module, 638
 mmpose.datasets module, 739
 mmpose.datasets.pipelines module, 808
 mmpose.datasets.pipelines.bottom_up_transform module, 813
 mmpose.datasets.pipelines.loading module, 808
 mmpose.datasets.pipelines.mesh_transform module, 816
 mmpose.datasets.pipelines.pose3d_transform module, 816
 mmpose.datasets.pipelines.shared_transform module, 808
 mmpose.datasets.pipelines.top_down_transform module, 811
 mmpose.models.backbones module, 645
 mmpose.models.detectors module, 679
 mmpose.models.heads module, 707
 mmpose.models.losses module, 726
 mmpose.models.misc module, 738
 mmpose.models.necks module, 675
 mmpose.utils module, 821
 MobileNetV2 (class in mmpose.models.backbones), 653
 MobileNetV3 (class in mmpose.models.backbones), 654
 ModelSetEpochHook (class in mmpose.core.utils), 638
 module
 mmpose.apis, 591
 mmpose.core.evaluation, 627
 mmpose.core.fp16, 636
 mmpose.core.post_processing, 639
 mmpose.core.utils, 638
 mmpose.datasets, 739
 mmpose.datasets.pipelines, 808
 mmpose.datasets.pipelines.bottom_up_transform, 813
 mmpose.datasets.pipelines.loading, 808
 mmpose.datasets.pipelines.mesh_transform, 816
 mmpose.datasets.pipelines.pose3d_transform, 816
 mmpose.datasets.pipelines.shared_transform, 808
 mmpose.datasets.pipelines.top_down_transform, 811
 mmpose.models.backbones, 645
 mmpose.models.detectors, 679
 mmpose.models.heads, 707
 mmpose.models.losses, 726
 mmpose.models.misc, 738
 mmpose.models.necks, 675
 mmpose.utils, 821
 MonitorNode (class in mmpose.apis.webcam.nodes), 616
 MoshDataset (class in mmpose.datasets), 772
 MPJPELoss (class in mmpose.models.losses), 730
 MSELoss (class in mmpose.models.losses), 731
 MSPN (class in mmpose.models.backbones), 652
 MTA (class in mmpose.models.necks), 676
 multi_gpu_test() (in module mmpose.apis), 596
 MultiItemProcess (class in mmpose.datasets.pipelines.shared_transform), 810
 multilabel_classification_accuracy() (in module mmpose.core.evaluation), 634

- MultiLossFactory (class in *mmpose.models.losses*), 732
- MultiModalSSAHead (class in *mmpose.models.heads*), 716
- MultiTask (class in *mmpose.models.detectors*), 693
- MultitaskGatherTarget (class in *mmpose.datasets.pipelines.shared_transform*), 810
- ## N
- nearby_joints_nms() (in module *mmpose.core.post_processing*), 642
- Node (class in *mmpose.apis.webcam.nodes*), 601
- norm1 (*mmpose.models.backbones.HRNet* property), 649
- norm1 (*mmpose.models.backbones.ResNet* property), 662
- norm1 (*mmpose.models.backbones.ViPNAS_ResNet* property), 674
- norm2 (*mmpose.models.backbones.HRNet* property), 649
- NormalizeJointCoordinate (class in *mmpose.datasets.pipelines.pose3d_transform*), 819
- NormalizeTensor (class in *mmpose.datasets.pipelines.shared_transform*), 810
- NoticeBoardNode (class in *mmpose.apis.webcam.nodes*), 611
- NVGestureDataset (class in *mmpose.datasets*), 772
- ## O
- ObjectAssignerNode (class in *mmpose.apis.webcam.nodes*), 615
- ObjectVisualizerNode (class in *mmpose.apis.webcam.nodes*), 610
- OffsetGenerator (class in *mmpose.datasets.pipelines.bottom_up_transform*), 815
- oks_iou() (in module *mmpose.core.post_processing*), 643
- oks_nms() (in module *mmpose.core.post_processing*), 643
- on_exit() (*mmpose.apis.webcam.nodes.Node* method), 602
- on_exit() (*mmpose.apis.webcam.nodes.RecorderNode* method), 618
- OneHand10KDataset (class in *mmpose.datasets*), 773
- ## P
- PAFGenerator (class in *mmpose.datasets.pipelines.bottom_up_transform*), 815
- PanopticDataset (class in *mmpose.datasets*), 774
- ParametricMesh (class in *mmpose.models.detectors*), 694
- PhotometricDistortion (class in *mmpose.datasets.pipelines.shared_transform*), 810
- pose_pck_accuracy() (in module *mmpose.core.evaluation*), 634
- PoseLifter (class in *mmpose.models.detectors*), 697
- PoseSequenceToTensor (class in *mmpose.datasets.pipelines.pose3d_transform*), 819
- PoseTrackerNode (class in *mmpose.apis.webcam.nodes*), 607
- PoseWarper (class in *mmpose.models.detectors*), 699
- PoseWarperNeck (class in *mmpose.models.necks*), 677
- post_dark_udp() (in module *mmpose.core.evaluation*), 635
- process() (*mmpose.apis.webcam.nodes.BaseVisualizerNode* method), 603
- process() (*mmpose.apis.webcam.nodes.DetectorNode* method), 605
- process() (*mmpose.apis.webcam.nodes.MonitorNode* method), 617
- process() (*mmpose.apis.webcam.nodes.Node* method), 602
- process() (*mmpose.apis.webcam.nodes.ObjectAssignerNode* method), 615
- process() (*mmpose.apis.webcam.nodes.PoseTrackerNode* method), 609
- process() (*mmpose.apis.webcam.nodes.RecorderNode* method), 618
- process() (*mmpose.apis.webcam.nodes.TopDownPoseEstimatorNode* method), 607
- process_mmdet_results() (in module *mmpose.apis*), 597
- project_joints() (*mmpose.models.losses.SemiSupervisionLoss* static method), 734
- project_points() (*mmpose.models.losses.MeshLoss* method), 732
- PyramidVisionTransformer (class in *mmpose.models.backbones*), 655
- PyramidVisionTransformerV2 (class in *mmpose.models.backbones*), 656
- ## Q
- quantize_float() (*mmpose.models.backbones.RegNet* static method), 659
- ## R
- RecorderNode (class in *mmpose.apis.webcam.nodes*), 617
- register_event() (*mmpose.apis.webcam.nodes.Node* method), 602
- register_input_buffer() (*mmpose.apis.webcam.nodes.Node* method),

[602](#)
[register_output_buffer\(\)](#) (*mm-pose.apis.webcam.nodes.Node* [method](#)), [602](#)
[RegNet](#) (*class* in *mmpose.models.backbones*), [657](#)
[RelativeJointRandomFlip](#) (*class* in *mm-pose.datasets.pipelines.pose3d_transform*), [820](#)
[RenameKeys](#) (*class* in *mm-pose.datasets.pipelines.shared_transform*), [811](#)
[report\(\)](#) (*mmpose.utils.StopWatch* [method](#)), [821](#)
[report_strings\(\)](#) (*mmpose.utils.StopWatch* [method](#)), [821](#)
[ResNeSt](#) (*class* in *mmpose.models.backbones*), [659](#)
[ResNet](#) (*class* in *mmpose.models.backbones*), [661](#)
[ResNetV1d](#) (*class* in *mmpose.models.backbones*), [662](#)
[ResNeXt](#) (*class* in *mmpose.models.backbones*), [660](#)
[RLELoss](#) (*class* in *mmpose.models.losses*), [733](#)
[rotate_point\(\)](#) (*in* *module* *mm-pose.core.post_processing*), [643](#)
[RSN](#) (*class* in *mmpose.models.backbones*), [656](#)
[run\(\)](#) (*mmpose.apis.webcam.nodes.Node* [method](#)), [602](#)

S

[SCNet](#) (*class* in *mmpose.models.backbones*), [662](#)
[screen_matting\(\)](#) (*in* *module* *mm-pose.apis.webcam.utils*), [624](#)
[SemiSupervisionLoss](#) (*class* in *mm-pose.models.losses*), [734](#)
[SEResNet](#) (*class* in *mmpose.models.backbones*), [664](#)
[SEResNeXt](#) (*class* in *mmpose.models.backbones*), [663](#)
[set_executor\(\)](#) (*mmpose.apis.webcam.nodes.Node* [method](#)), [603](#)
[set_executor\(\)](#) (*mm-pose.apis.webcam.nodes.ObjectAssignerNode* [method](#)), [616](#)
[set_train_epoch\(\)](#) (*mm-pose.models.detectors.GestureRecognizer* [method](#)), [691](#)
[set_train_epoch\(\)](#) (*mm-pose.models.heads.MultiModalSSAHead* [method](#)), [717](#)
[setup_multi_processes\(\)](#) (*in* *module* *mmpose.utils*), [822](#)
[show_result\(\)](#) (*mmpose.models.detectors.AssociativeEmbedding* [method](#)), [681](#)
[show_result\(\)](#) (*mmpose.models.detectors.CID* [method](#)), [684](#)
[show_result\(\)](#) (*mmpose.models.detectors.DetectAndRegress* [method](#)), [687](#)
[show_result\(\)](#) (*mmpose.models.detectors.DisentangledKeypointRegression* [method](#)), [690](#)
[show_result\(\)](#) (*mmpose.models.detectors.GestureRecognizer* [method](#)), [691](#)
[show_result\(\)](#) (*mmpose.models.detectors.Interhand3D* [method](#)), [692](#)
[show_result\(\)](#) (*mmpose.models.detectors.ParametricMesh* [method](#)), [696](#)
[show_result\(\)](#) (*mmpose.models.detectors.PoseLifter* [method](#)), [698](#)
[show_result\(\)](#) (*mmpose.models.detectors.TopDown* [method](#)), [702](#)
[show_result\(\)](#) (*mmpose.models.detectors.VoxelCenterDetector* [method](#)), [705](#)
[show_result\(\)](#) (*mmpose.models.detectors.VoxelSinglePose* [method](#)), [707](#)
[ShuffleNetV1](#) (*class* in *mmpose.models.backbones*), [666](#)
[ShuffleNetV2](#) (*class* in *mmpose.models.backbones*), [667](#)
[single_gpu_test\(\)](#) (*in* *module* *mmpose.apis*), [597](#)
[singleTagLoss\(\)](#) (*mmpose.models.losses.AELoss* [method](#)), [727](#)
[smooth\(\)](#) (*mmpose.core.post_processing.Smoother* [method](#)), [640](#)
[smooth_l1_loss\(\)](#) (*mm-pose.models.losses.SoftWeightSmoothL1Loss* [static method](#)), [736](#)
[Smoother](#) (*class* in *mmpose.core.post_processing*), [639](#)
[SmoothL1Loss](#) (*class* in *mmpose.models.losses*), [735](#)
[smp1_losses\(\)](#) (*mmpose.models.losses.MeshLoss* [method](#)), [732](#)
[soft oks nms\(\)](#) (*in* *module* *mm-pose.core.post_processing*), [644](#)
[SoftWeightSmoothL1Loss](#) (*class* in *mm-pose.models.losses*), [735](#)
[SoftWingLoss](#) (*class* in *mmpose.models.losses*), [736](#)
[split_ae_outputs\(\)](#) (*in* *module* *mm-pose.core.evaluation*), [635](#)
[StopWatch](#) (*class* in *mmpose.utils*), [821](#)
[SunglassesEffectNode](#) (*class* in *mm-pose.apis.webcam.nodes*), [612](#)
[SwinTransformer](#) (*class* in *mmpose.models.backbones*), [668](#)
[sync_random_seed\(\)](#) (*in* *module* *mmpose.core.utils*), [639](#)

T

[TCFormer](#) (*class* in *mmpose.models.backbones*), [669](#)
[TCN](#) (*class* in *mmpose.models.backbones*), [670](#)
[TemporalRegressionHead](#) (*class* in *mm-pose.models.heads*), [717](#)
[timeit\(\)](#) (*mmpose.utils.StopWatch* [method](#)), [821](#)
[TopDown](#) (*class* in *mmpose.models.detectors*), [701](#)
[TopDownAffine](#) (*class* in *mm-pose.datasets.pipelines.top_down_transform*), [811](#)
[TopDownAicDataset](#) (*class* in *mmpose.datasets*), [775](#)

- [TopDownCocoDataset](#) (class in *mmpose.datasets*), 776
[TopDownCocoWholeBodyDataset](#) (class in *mmpose.datasets*), 777
[TopDownCrowdPoseDataset](#) (class in *mmpose.datasets*), 778
[TopDownFreiHandDataset](#) (class in *mmpose.datasets*), 779
[TopDownGenerateTarget](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 811
[TopDownGenerateTargetRegression](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 812
[TopDownGetBboxCenterScale](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 812
[TopDownGetRandomScaleRotation](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 812
[TopDownH36MDataset](#) (class in *mmpose.datasets*), 779
[TopDownHalfBodyTransform](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 812
[TopdownHeatmapBaseHead](#) (class in *mmpose.models.heads*), 719
[TopdownHeatmapMSMUHead](#) (class in *mmpose.models.heads*), 720
[TopdownHeatmapMultiStageHead](#) (class in *mmpose.models.heads*), 721
[TopdownHeatmapSimpleHead](#) (class in *mmpose.models.heads*), 723
[TopDownJhmdbDataset](#) (class in *mmpose.datasets*), 780
[TopDownMhpDataset](#) (class in *mmpose.datasets*), 781
[TopDownMpiiDataset](#) (class in *mmpose.datasets*), 782
[TopDownMpiiTrbDataset](#) (class in *mmpose.datasets*), 784
[TopDownOCHumanDataset](#) (class in *mmpose.datasets*), 785
[TopDownOneHand10KDataset](#) (class in *mmpose.datasets*), 786
[TopDownPanopticDataset](#) (class in *mmpose.datasets*), 786
[TopDownPoseEstimatorNode](#) (class in *mmpose.apis.webcam.nodes*), 606
[TopDownPoseTrack18Dataset](#) (class in *mmpose.datasets*), 786
[TopDownPoseTrack18VideoDataset](#) (class in *mmpose.datasets*), 788
[TopDownRandomFlip](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 812
[TopDownRandomShiftBboxCenter](#) (class in *mmpose.datasets.pipelines.top_down_transform*), 813
[ToTensor](#) (class in *mmpose.datasets.pipelines.shared_transform*), 811
[train\(\)](#) (*mmpose.models.backbones.HRNet* method), 649
[train\(\)](#) (*mmpose.models.backbones.LiteHRNet* method), 652
[train\(\)](#) (*mmpose.models.backbones.MobileNetV2* method), 654
[train\(\)](#) (*mmpose.models.backbones.MobileNetV3* method), 655
[train\(\)](#) (*mmpose.models.backbones.ResNet* method), 662
[train\(\)](#) (*mmpose.models.backbones.ShuffleNetV1* method), 666
[train\(\)](#) (*mmpose.models.backbones.ShuffleNetV2* method), 667
[train\(\)](#) (*mmpose.models.backbones.SwinTransformer* method), 669
[train\(\)](#) (*mmpose.models.backbones.VGG* method), 672
[train\(\)](#) (*mmpose.models.backbones.ViPNAS_MobileNetV3* method), 673
[train\(\)](#) (*mmpose.models.backbones.ViPNAS_ResNet* method), 675
[train\(\)](#) (*mmpose.models.detectors.DetectAndRegress* method), 687
[train\(\)](#) (*mmpose.models.necks.PoseWarperNeck* method), 678
[train_model\(\)](#) (in module *mmpose.apis*), 597
[train_step\(\)](#) (*mmpose.models.detectors.DetectAndRegress* method), 687
[train_step\(\)](#) (*mmpose.models.detectors.ParametricMesh* method), 696
[transform_preds\(\)](#) (in module *mmpose.core.post_processing*), 644
V
[V2VNet](#) (class in *mmpose.models.backbones*), 671
[val_step\(\)](#) (*mmpose.models.detectors.ParametricMesh* method), 697
[vertex_loss\(\)](#) (*mmpose.models.losses.MeshLoss* method), 732
[VGG](#) (class in *mmpose.models.backbones*), 671
[VideoEndingMessage](#) (class in *mmpose.apis.webcam.utils*), 620
[ViPNAS_MobileNetV3](#) (class in *mmpose.models.backbones*), 672
[ViPNAS_ResNet](#) (class in *mmpose.models.backbones*), 673
[ViPNASHeatmapSimpleHead](#) (class in *mmpose.models.heads*), 724
[vis_3d_mesh_result\(\)](#) (in module *mmpose.apis*), 597
[vis_3d_pose_result\(\)](#) (in module *mmpose.apis*), 597
[vis_pose_result\(\)](#) (in module *mmpose.apis*), 598

`vis_pose_tracking_result()` (in module `mm-pose.apis`), [598](#)

`VoxelCenterDetector` (class in `mm-pose.models.detectors`), [703](#)

`VoxelSinglePose` (class in `mmpose.models.detectors`), [705](#)

W

`warp_affine_joints()` (in module `mm-pose.core.post_processing`), [644](#)

`WebcamExecutor` (class in `mmpose.apis.webcam`), [599](#)

`WeightNormClipHook` (class in `mmpose.core.utils`), [638](#)

`WingLoss` (class in `mmpose.models.losses`), [737](#)

`with_keypoint` (`mmpose.models.detectors.AssociativeEmbedding` property), [681](#)

`with_keypoint` (`mmpose.models.detectors.CID` property), [685](#)

`with_keypoint` (`mmpose.models.detectors.DisentangledKeypointRegressor` property), [690](#)

`with_keypoint` (`mmpose.models.detectors.PoseLifter` property), [699](#)

`with_keypoint` (`mmpose.models.detectors.TopDown` property), [703](#)

`with_neck` (`mmpose.models.detectors.PoseLifter` property), [699](#)

`with_neck` (`mmpose.models.detectors.TopDown` property), [703](#)

`with_necks` (`mmpose.models.detectors.MultiTask` property), [694](#)

`with_traj` (`mmpose.models.detectors.PoseLifter` property), [699](#)

`with_traj_backbone` (`mmpose.models.detectors.PoseLifter` property), [699](#)

`with_traj_neck` (`mmpose.models.detectors.PoseLifter` property), [699](#)

`wrap_fp16_model()` (in module `mmpose.core.fp16`), [638](#)