Joint COCO and Mapillary Workshop at ICCV 2019: COCO 2019 Keypoint Detection Task Challenge Track

Technical Report: Deep Structured Fusion Network via Progressive Learning for Human Pose Estimation

Jia Wei*
Netease Game AI Lab
weijia@corp.netease.com

Yanjun Li*
Netease Game AI Lab
liyanjun@corp.netease.com

Abstract

We use a two stage top-down method. In the first stage, we use Hybrid Task Cascade to detect human bounding boxes. In the second stage, we use our Structured Fusion Network with Deconvolution Head and two-stage network to detect human keypoints. Finally we use Refinement Network to refine the results got by using ensemble method(5 models). We train models using extra data(AI-Challenger dataset). Our best single model has 0.764 of mAP on COCO test-dev dataset, and final result has 0.775 of mAP on COCO test-dev dataset and 0.751 of mAP on COCO test-challenge dataset.

1. Introduction

2D human pose estimation problem has been a fundamental research topic in computer vision. The goal is to recognize and locate human keypoints (e.g. nose, eye, ankle, etc.). Recently, the problem of human pose estimation has been rapid improved by using the deep convolution neural networks[5]. This technical report introduces two improvements in comparison to existing networks for pose estimations. (i)Our approach fuses the rough human keypoints information into the pose estimation network. (ii)we use a multi-stage architecture and aggregate features across different stages to strengthen the information flow,in order to estimate more accurate pose progressively.

With the above improvements,our approach is slightly better than HRNet[9]. On COCO keypoint benchmark,our single model achieve 75.7 average precison(AP) on test-dev.With additional data from AI Challenger[11] for training and network ensemble, our approach can obtain an AP of 77.5.

*The two authors contribute equally to this work.

2. Approach

We adopt the top-down approach in two steps. In the first stage, we use Hybrid Task Cascade to detect human bounding boxes. In the second stage, we use our Structured Fusion Network with Deconvolution Head to detect human keypoints. The following sections elaborate the network designs.

2.1. Fuse Structure Information

Most existing networks take the origin image as input,without additional information. In our network,we fuse the rough human keypoints information and the image together. Firstly,we train a simple subnetwork for pose estimation. The feature map outputed by the last layer of the subnetwork indicates the rough keypoints locations of the input image. Then,the feature map will be fused into the second subnetwork in various resolutions. With the help of the rough keypoints location,the network can pay more attention to the specific area in the image. The two subnetwork will be optimized together.

2.2. Progressive Learning

Many recent methods use multi-stage architure and we follow the common practice. HRNet[9] is adopted as the backbone network. We aggregate features across different stages to strengthen the information flow. The feature map in the previsou stage will be added into the next stage in different resolutions. With this design,the current stage can take advantage of prior information. The last layer feature maps in different stages will be optimized together during training procedures.

3. Experiments

3.1. COCO Keypoint Detection

Dataset and Evaluation Protocol. *The COCO dataset[6] contains more than 200,000 images and 250,000 person instances labeled with keypoints. We train our model on COCO train2017 dataset. We evaluate our model on the val2017 and test2017 set, containing 5,000 images and 40K images, respectively. For each person, ground truth keypoints have the form* $[x_1, y_1, v_1, ..., x_k, y_k, v_k]$*, where x,y are the keypoint locations and v is a visibility flag defined as v=0: not labeled, v=1: labeled but not visible, and v=2: labeled and visible. OKS-based mAP (AP for short) is used as our evaluation metric[6].*

Training. *The network is trained on 8 NVIDIA GTX 1080Ti GPUs with Adam optimize. The base learning rate is set as 1e-3. The mini-batch size is set as 16 per GPUs.*

Each human detection box will be extended in height or width to a fixed aspect ratio and then cropped from the image, which is resized to a fixed size 384×288*. Then we will randomly go through a series of data augmentation operations including flipping, rotation and scaling. As for rotation, the rotation range is* $-45° \sim 45°$*, and scaling range is* $0.65 \sim 1.35$*.*

Testing. *We use a two-stage top-down paradigm similar [10, 12] as is used: detect a single person instance from image using a person detector, and then predict keypoints with our model. Due to lack of time, We use an object detector*[1] *[2] training on COCO dataset instead of person detector, which may cause AP to decrease. Following the same strategy as [9], we average the predicted heatmaps of original image with results of corresponding flipped image. Each keypoint location is predicted by adjusting the highest heatvalue location with a quarter offset in the direction from the highest response to the second highest response.*

3.2. Ablation Study

In this section, we provide an ablation analysis in our approach.

3.2.1 Influence of Human Detector

We use an object detector which detected 80 object categories. It has 50.7 box AP on COCO dataset. Unfortunately, it got worse performance for keypoints than other single person detector. To evaluate its influence on the final pose estimation accuracy, we test on COCO val2017 using the detect results in [9] and the "oracle detector" using ground truth boxes.

[1] https://github.com/open-mmlab/mmdetection

Pose estimation performance using different detector is reported in Table 1. Obviously, the accuracy of the SFNet using the object detection on the COCO val2017 set is 0.8 higher than that using the person detector. However, with a object detector, the human pose estimation accuracy is slightly reduced on COCO test-dev2017. The influence of detector is quite obvious and a little wired. It shows that on COCO val2019 dataset we got 0.8 gain but got 0.2 reduction on COCO test-dev2017 dataset.

Detector	val2017 AP	test-dev2017 AP
object detector	0.782	0.764
person detector	0.774	0.766
gronud truth	0.797	-

Table 1: Results of SFNet using three detectors on COCO val2017 and test-dev2017.

3.2.2 Results on the Test Set

Table 2 reports the pose estimation performances of our approach and the existing state-of-the-art approaches. Our approach is slightly better than HRNet[9]. With additional data from AI Challenger[11] for training, our approach can obtain an AP of 77.5. From Table 3, it is clear that our approach obtains 75.1 AP on the test-challenge dataset and shows its slightly superiority over HRNet methods.

4. Conclusion

In this work, we propose a SFNet-PL approach to perform human pose estimation. Our implementation shows that our approach is slightly better than HRNet[9]. We first use an object detector to detect person from images, and then use SFNet-PL to predict the localization of person keypoints. Due to lack of time, we have to use an object detector[2] to detect the location of person from images, which lead to the reduction of AP on COCO test set.

References

[1] *Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In* CVPR, *2017. 3*

[2] *Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In* IEEE Conference on Computer Vision and Pattern Recognition, *2019. 2*

[3] *Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In* CVPR, *2018. 3*

[4] *Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In* ICCV, *2017. 3*

Method	Backbone	Input Size	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
CMU Pose [1]	-	-	61.8	84.9	67.5	57.1	68.2	66.5	87.2	71.8	60.6	74.6
Mask R-CNN [4]	Res-50-FPN	-	63.1	87.3	68.7	57.8	71.4	-	-	-	-	-
G-RMI [8]	Res-152	353×257	64.9	85.5	71.3	62.3	70.0	69.7	88.7	75.5	64.4	77.1
AE [7]	-	512×512	65.5	86.8	72.3	60.6	72.6	70.2	89.5	76.0	64.6	78.1
CPN [3]	Res-Inception	384×288	72.1	91.4	80.0	68.7	77.2	78.5	95.1	85.3	74.2	84.3
Simple Base [12]	Res-152	384×288	73.7	91.9	81.1	70.3	80.0	79.0	-	-	-	-
HRNet [9]	HRNet-W48	384×288	75.5	92.5	83.3	71.9	81.5	80.5	-	-	-	-
CPN+ [3]	Res-Inception	384×288	73.0	91.7	80.9	69.5	78.1	79.0	95.1	85.9	74.8	84.6
Simple Base+* [12]	Res-152	384×288	76.5	92.4	84.0	73.0	82.7	81.5	95.8	88.2	77.4	87.2
HRNet* [9]	HRNet-W48	384×288	77.0	92.7	84.5	73.4	83.1	82.0	-	-	-	-
Ours(SFNet-PL+*)	-	384×288	**77.5**	**92.7**	**84.8**	**74.3**	**83.1**	**82.3**	**95.9**	**88.8**	**78.2**	**87.9**

Table 2: Comparisons of results on COCO test-dev dataset. "+" indicates using an ensemble model and "*" means using external data.

Method	Backbone	Input Size	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{50}	AR^{75}	AR^{M}	AR^{L}
Mask R-CNN* [4]	ResX-101-FPN	-	68.9	89.2	75.2	63.7	76.8	75.4	93.2	81.2	70.2	82.6
G-RMI* [8]	Res-152	353×257	69.1	85.9	75.2	66.0	74.5	75.1	90.7	80.7	69.7	82.4
CPN+ [3]	Res-Inception	384×288	72.1	90.5	78.9	67.9	78.1	78.7	94.7	84.8	74.3	84.7
Sea Monsters+*	-	-	74.1	90.6	80.4	68.5	82.1	79.5	94.4	85.1	74.1	86.8
Simple Base+* [12]	Res-152	384×288	74.5	90.9	80.8	69.5	82.9	80.5	95.1	86.3	75.3	87.5
Ours(SFNet-PL+*)	-	384×288	**75.1**	**91.2**	**81.2**	**70.4**	**82.4**	**80.5**	**94.9**	**86.1**	**75.3**	**87.4**

Table 3: Comparisons of results on COCO test-challenge dataset. "+" means using an ensemble model and "*" means using external data.

[5] *Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In* IEEE Conference on Computer Vision and Pattern Recognition*, 2016.* 1

[6] *Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In* ECCV*, 2014.* 2

[7] *Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In* NIPS*, 2017.* 3

[8] *George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In* CVPR*, 2017.* 3

[9] *Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In* CVPR*, 2019.* 1, 2, 3

[10] *Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In* ECCV*, 2018.* 2

[11] *Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, Yizhou Wang, and Yonggang Wang. AI challenger : A large-scale dataset for going deeper in image understanding.* CoRR*, abs/1711.06475, 2017.* 1, 2

[12] *Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In* ECCV*, 2018.* 2, 3