Multi-Person Pose Estimation for PoseTrack with Enhanced Part Affinity Fields

Xiangyu Zhu, Yingying Jiang, and Zhenbo Luo
Machine Learning Lab
Samsung R&D Institute of China, Beijing
Beijing, China
{xiangyu.zhu, yy.jiang} @samsung.com

***Abstract*: This paper is a description for method we adopted in the competition of "PoseTrack, ICCV 2017 workshop" [1]. We presents an improved approach based on Part Affinity Fields (PAFs) [2]. To achieve a better performance on PoseTrack benchmark, several modifications are proposed, including pre-training model on COCO [3], rethinking the network structure and redundant PAFs. As a result, the framework obtains a significant improvement comparing to baseline methods. Moreover, inspired by semantic segmentation, we conduct some experiments using the hole algorithm and DenseNet, which achieves a desirable performance. Our submission achieves 72.5% mAP on PoseTrack validation dataset and 68.3% on Posetrack benchmark.**

I. INTRODUCTION

Human pose estimation attracts increasing attentions, not only from researchers, but also from many corporations. One of the main applications is to understand human activity and interactions, which is mentioned frequently in existing literature. But now it comes to some specific scenarios. For example, self-driving car companies use it to understand the pedestrian's action and intention. Elder care robot can detect the fall down events by analyzing user's body pose. Some companies have already developed a prototype or demo using human pose estimation.

In order to apply this technology to self-driving car or care robot, we need to address some challenging problems, such as human pose estimation for multi-person, due to multi-person interaction and occlusion. PoseTrack [1] dataset provides numerous images clipped from videos. In the images, multi-person interact with each other. This benchmark presents the common scene in daily life, and would act as a persuasive index for algorithm robustness.

In this work, we present an improved approach based on Cao's [2] framework, which is the champion of COCO 2016 keypoints challenge [3], and discuss some potential weakness of this method. First, to enjoy the benefits of more training data, we pre-train the model on COCO dataset. Second, we extend the original Part Affinity Fields (PAFs) mechanism to redundant PAFs, which reveals an essential defect of PAFs. Third, by rethinking the network structure itself, we find out some slight modifications that can lead to remarkable improvement. The submission is implemented with these three modifications.

Additionally, inspired by semantic segmentation, we design some experiments that exploit semantic segmentation framework, such as Deeplab [4] and SDN [5]. We also have tried DenseNet [6] and the holealgorithm. But limited to the deadline, we have not used this framework as the final version. Thus, the submission result is unrelated to these experiments. Although we have not obtained an out-performing result, some conclusions might be helpful for future work of multi-person pose estimation.

II. RELATED WORK

We briefly review the two categories of multi-person pose estimation approaches. Then the advantages and disadvantages of them are discussed.

Most of the multi-person pose estimation approaches can be categorized into top-down approach and bottom-up approach. Top-down approachis the most common method, which uses person detector and performs single person estimation for each individual. Some methods [1, 7] concatenate detector and person estimation in sequence, and others [8, 9] predict person bounding box and joints simultaneously, in a unified network. Bottom-up approach [2, 10] first predicts individual body joints and then groups them into persons. Instead of person detector, these methods dig out some inner relation between individual joints, such as middle points and limb vectors.

The main advantage of top-down approach is obvious, it exploit high accuracy single person estimation. However, it heavily depends on the reliability of person detector. And once the number of person increases, the computation increases linearly. It will become extremely slow when numerous persons exist. On the contrary, bottom-up approach keep computation constant in this case. Whereas joints relation might fail to use context information and might not be as reliable as person detector. After all, person detector or pedestrian detection is becoming increasingly accurate, due to the breakthroughs in classification and detection.

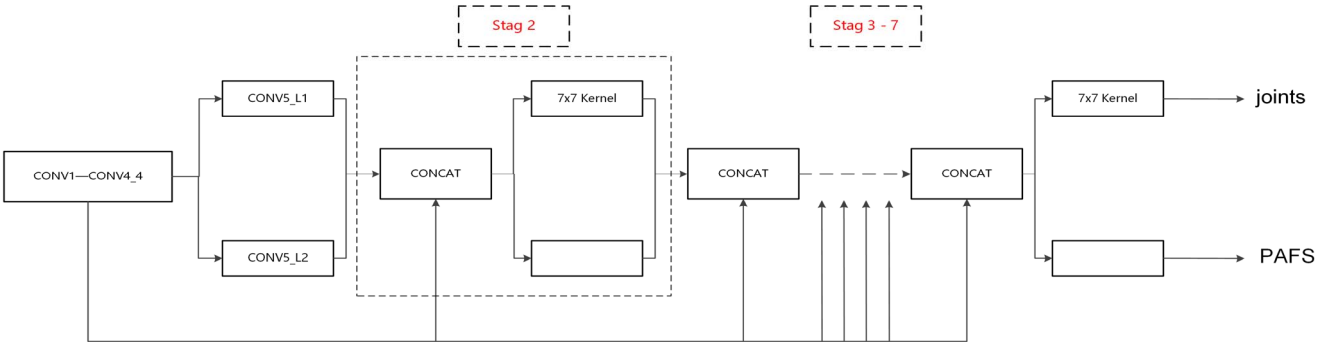


Figure 1 Overview of our framework

III. METHOD

Figure 1 depicts the overview of our framework. Comparing with the baseline method, we make a few modifications, which are illustrated in Section B and Section C. We evaluate baseline method and modified framework on 100 images sampled from validation set. All the performances mentioned in this section are tested on the 100 images. In Section D, we carry out some experiments referring to semantic segmentation frameworks, which is not included in submission version.

A. Baselinemethodpre-trained on COCO

Our approach is mainly based on Cao's [2] framework, which is categorized as a bottom-up method. Unlike top-down method, computation and running time stay almost constant when the number of person increases.

The baseline framework is designed as multi-task network, predicting joints and PAFs simultaneously. The key insight of this work is to formulate limb vectors as a dense prediction task, similar to joints prediction. The author modified the data input layer, making it able to generate mask maps as ground-truth.

Another highlight of this work is using big kernel convolution layers in refinement stages [11]. In most networks, they only use a kernel size at 1 or 3. But in the baseline method, it adopts a big kernel size up to 7. Big kernel convolution layers extend effective receptive fields to 400 pixel, which make it possible to using context information. On the other hand, the two tasks enjoy mutual benefits from each other, either joints and PAFs task can benefit each other.

The following paragraphs illustrate the training strategy with baseline method. The baseline model is pre-trained on COCO and then finetuned on PoseTrack dataset. Table 1 shows the performance of every training step on validation dataset.

Train on Posetrack. Firstly, we train a baseline on PoseTrack dataset. We use the caffe implementations, which is an official project provided by the author. The performance of the baseline on validation set is 58% (mAP), about 15% (mAP) lower than final version.

Train on COCO. Secondly, we train the model on COCO dataset. However, the keypoints annotated in COCO are not completely corresponding to PoseTrack's. Head-top is not annotated in COCO dataset while eyes and ears are annotated. As a result,the model pre-trained on COCO does not predict head-top keypoint, leading to an extremely low score on head. Even though one keypoint is absent, baseline pre-trained on COCO achieves a much higher performance, about 65% on validation set.

Finetune on Posetrack. Finally, to compensate the absence of head-top keypoint, we fine-tune the pre-trained model on PoseTrack dataset. Empirically, we find the fine-tune process must be controlled strictly. At the first 1000 to 2000 iterations, the performance increases stably. However, with more fine-tune iterations, the performance decreases rapidly.

Table 1 Baseline performance while training on different dataset

Dataset	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP
Posetrack	77.2	67.1	56.4	43.0	51.8	44.7	37.2	55.5
COCO	44.9	84.2	72.8	58.1	75.5	62.2	54.5	63.3
COCO +Posetrack	85.4	84.0	71.2	58.0	72.6	61.6	55.0	70.7

B. Redundant Part Affinity Fields

The main function of Part Affinity Fields is to group the discrete joints into individual persons. It detects the vector or connection between joints. However, this mechanism has a fatal defect.PAFs connect N points with (N-1) lines, this is the minimum connection number to associate all the joints, shown in figure 1 (a). That means, in order to obtain a whole human skeleton, it is needed to make every connection prediction and joint prediction exactly correct. This characteristic weakens the robustness of the mechanism.

Moreover, since the joints are concatenated one by one in tree structure, once the parentlink is broken, the child connection and joints will be abandoned, even though detected correctly. For example, there are three joints concatenated in sequence, from shoulder, elbow to wrist. The connection between shoulder and elbow is missing. Thus, this algorithm fails to group wrist joint to individual person.

To address this problem, we develop a redundant Part Affinity Fields (PAFs). We increase the connection number between joints, making the connection redundant. Figure 2

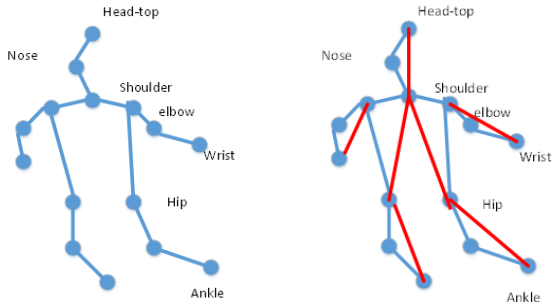
shows the comparison between original PAFs and our redundant PAFs. By adding these redundant lines, fault-tolerate-rate increases. It is not necessary to detect every connection correctly, since some joint involves multiply connections. The structure between joints is transformed from tree to graph.

Limited to the time, we conduct the ablation experiment when pre-training on COCO dataset, shown in Table 2. We can observe a significant improvement at wrist, hip and ankle, about 1.5-2%, which proves the effectiveness of redundant PAFs. The ablation experiment uses the same model file, the only difference is using redundant PAFs information or not.

Apparently, this redundant PAFs is just a naïvecompensation for PAFs. The main contribution of this section is that it reveals the main weakness of PAFs.

Table 2 Ablation experiment for redundant PAFs

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP
Without redundant PAFs	45.3	84.5	73.3	59.4	75.5	62.9	55.3	63.8
With redundant PAFs	45.6	84.4	73.3	**60.8**	**77.1**	63.1	**57.2**	**64.6**



(a) (b)

Figure.2. Comparison between PAFs and redundant PAFs. (a) Original PAFs (b) Redundant PAFs, red lines indicate redundant PAFs .

C. *Rethinking the network structure*

Feature extraction. Baseline method uses the first 10 layers of VGG-19 [12] and re-designs the following layers.There is a puzzle that they did not use the whole conv4 of VGG-16, instead they used the feature maps from intermediate layers. Huang et al.[13] have carried out an ablation experiment to analyze the performance when adding intermediate classifier to each convolution layer. The performance drops rapidly at intermediate layers in VGG-like networks. Thus, we simply change the output feature maps from intermediate layer conv4_2 to the whole bank conv4. To our surprise, this slight modification contributes more than 2% improvement to performance. Ablation experiments are shown in Table 3.

More refinement stage. Using big kernel convolution layer is another highlight of baseline method, which expands the receptive fields to 400 pixels.With more refinement stages, we can observe a remarkable decrease of loss in training process. Loss of Stage 7 is lower than Stage 6. In this work, we adopt 7 stage framework instead of 6 stage framework. It contributes approximately 1% to performance. Ablation experiments are shown in Table 3.

Table 3 Effect of changing network structure

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP
Baseline	44.9	84.2	72.8	58.1	75.5	62.2	54.5	63.3
+whole conv4	46.1	85.5	75.2	61.0	77.3	65.6	59.9	65.8
+whole conv4 +stage 7	45.9	86.4	77.1	64.8	79.4	66.9	58.0	66.9

D. *Experiments inspired by semantic segmentation.*

Almost all the human pose estimation methods could be summarized as dense prediction approach. It outputs heatmaps that reflect joints in pixel level. Similar to human pose estimation, semantic segmentation is also formulated as dense prediction problem, every pixel in the image would be classified. This similarity shows a strong connection between human pose estimation and semantic segmentation. In fact, some approaches [8, 9] could accomplish both of these two tasks and achieve the-state-of-art performance on COCO keypoint benchmark. Thus, we assume that pose estimation network can benefit from algorithm used in semantic segmentation. We carry out some experiment referring to semantic segmentation frameworks.

VGG-19 with the hole algorithm. Baseline method only uses the first four blocks of VGG-19 due to stride, while the last fifth block is abandoned. DeeperCut [14] and other dense prediction frameworks prove that stride 8 might be a proper choice for dense prediction. We refer to the framework of DeepLab-v2 [4], adopt the whole VGG-19 structure, using the hole algorithm to recover resolution from stride 16 to 8. We train the model on COCO dataset and test it on a subset of 2644 images derived from validation set. The result shows the performance on large targets increases while the performance on medium targets decreases. A reasonable hypothesis is that the hole algorithm increases the receptive fields as well as the big kernel refine stages. The effective receptive fields become too large to fit the small object.

DenseNet with the hole algorithm. Referring to SDN [5], we use DenseNet with the hole algorithm to replace VGG-19 in this experiments. DenseNet-121 is adopted as backbone, while conv5 bank is abandoned and conv4 is adapted to 8 stride using the hole algorithm. This framework turns out to be a stable and easily converging method. We only use two stages instead of six stages. The performance is 63% on validation set, comparing to 61% using two stages baseline method. However, when adding more refinement stages, the performance does not increase remarkably. A 6 stages implementation gets 64%, with a small margin to 2 stages method.

Analysis. Both of the experiments above adopt the hole algorithm, which seems contradict with big kernel refinement stages. We will analyze thisphenomena carefully in the future.

E. Training details

We use the official implementation from Cao [2] and initialize the VGG19 model from the ImageNet-pre-trained model. At first, we train our model on COCO dataset for 130000 iterations with lr=4e-5,stepsize= 40000, gamma = 0.333. This step costs 2 days using two P40 GPU. Then, we finetune the COCO pre-trained model on PoseTrack dataset for 3000 iterations, with lr=1e-5, stepsize=1000, gamma = 0.333. In finetuning process, we froze the weights of feature extraction layers, from conv1-conv4.

IV. FINNAL PERFORMANCE ON POSETRACK

We evaluate our method on the whole validation set and subset of testset. All the results in Table 4 and Table 5 are obtained from the evaluation server.

While our method performs well on head and shoulder, its performance on ankle and wrist is not so good. The reason is that ankle and wrist are easy to be hidden in multi-person pose estimation.

Table 2 Performance on PoseTrack validation set

	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP
Ours	83.8	84.9	76.2	64.0	72.2	64.5	56.6	72.6

Table 3 Performance on subset of PoseTrack testset

	Hea	Sho	Elb	Wri	Hip	Kne	Ank	mAP
Ours	79.5	78.7	71.8	62.0	65.9	60.3	54.3	68.3

V. CONCLUSIONS

In this paper, we present an improved framework based on PAFsto achieve better performance on PoseTrack Challenge. We develop a redundant PAFs mechanism to compensate the defect of original PAFs. Additionally, we carry out some experiments referring to semantic segmentation frameworks. And our modifications on networks are proved to be quite effective.

The paper is also a description of the method we used in PoseTrack competition.

REFERENCES

[1] Iqbal U, Milan A, Gall J. PoseTrack: Joint Multi-Person Pose Estimation and Tracking[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017.

[2] Cao Z, Simon T, Wei S E, et al. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields[J]. 2016.

[3] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[J]. 2014, 8693:740-755.

[4] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):1-1.

[5] Fu J, Liu J, Wang Y, et al. Stacked Deconvolutional Network for Semantic Segmentation[J]. 2017.

[6] Huang G, Liu Z, Weinberger K Q, et al. Densely Connected Convolutional Networks[J]. 2016.

[7] Fang H, Xie S, Tai Y, et al. RMPE: Regional Multi-person Pose Estimation[J]. 2016.

[8] Newell A, Huang Z, Deng J. Associative Embedding: End-to-End Learning for Joint Detection and Grouping[J]. 2016.

[9] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[J]. 2017.

[10] Pishchulin L, Insafutdinov E, Tang S, et al. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation[J]. 2015, 2008(1):4929-4937.

[11] Wei S E, Ramakrishna V, Kanade T, et al. Convolutional Pose Machines[J]. 2016:4724-4732.

[12] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[13] Huang G, Chen D, Li T, et al. Multi-Scale Dense Convolutional Networks for Efficient Prediction[J]. 2017.

[14] Insafutdinov E, Pishchulin L, Andres B, et al. DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model[J]. 2016, 42(5):34-50.