Human Pose Estimation with Occlusion

Abstract

Human Pose Estimation (HPE) has been popular in the computer vision community. Various deep learning models have been proposed to achieve superior performance on HPE. However, if parts of the objects are occluded, their performances would degrade due to the loss of context and semantics. Towards this problem, this work proposes an artificial occlusion transformation to imitate in-the-wild occlusions. Its use is tested on three well-known HPE models reproduced by ourselves, i.e. *SimpleBaseline, HRNet, and ViTPose. We first show how their performances have been affected when presented with occluded images. Experiments were then conducted to investigate the optimal occlusion settings. Finally, we concluded that fine-tuning images with occlusions could boost the robustness of the model.*

1. Introduction

Human Pose Estimation (HPE) has attracted much attention in the computer vision community. It aims at capturing the critical points of human body parts and their connectivity from images and videos. HPE can be applied to a lot of down-stream tasks, including Human Re-identification [3], Human Pose Tracking [17], Human Action Recognition [8]. The general motion information it captures can be utilized in Human-Computer Interaction, healthcare and augmented reality, and so on [6].

However, there are lots of challenges in HPE. To capture the body pose, all the key body points belonging to the object need to be detected accurately. But the distance from the cameras to objects varies [15]. Some joints are occluded by environments or other objects. Apart from those, the data quality of cameras would be influenced by the weather and the light conditions. Moreover, when there is more than one person, different key points need to be assigned to corresponding objects accurately. Also, the key joints can be articulated in various ways. When the number of key points goes large, all the possible combination forms a large search space. Besides, for 3D HPE, projecting from 2D to 3D introduces another error source.

Towards those problems, countless effort has been done. To describe the human body, the kinematics model, planar model, and volumetric model have been proposed [22]. From the problem scenarios perspective, existing methods can be further divided into 2D and 3D HPE, single-person and multi-person. For the single-person problem, methods can be divided into regression-based methods and body-part detection methods. For the multi-person problem, top-down/bottom-up pipelines have been widely investigated. The top-down pipelines detect and separate objects first and then detect key points for each person. The bottom-up pipelines detect all the key points at once and then assign them to each person. This paper aims at exploring the 2D single-person HPE.

With the proliferation of machine learning techniques and the advancement of computational resources, deep learning models have been widely used to solve the 2D single-person HPE problem. While early works in human pose estimates go back to 2001 [12], with the exponential scale of computing over the last two decades, sophisticated machine learning algorithms have taken the lead. Convolutional neural networks such as *Simple Baselines for Human Pose Estimation and Tracking* [18] with a ResNet-152 neural network reaches up to 73.7% average precision, with the best work, *DARK+extra* [4] (HRNet-W48 model) achieving 77.4% AP [4] on the COCO dataset. Transformer models such as ViTPose+ [19], based on the ViTAE-G model reach up to 81.1% AP which outperforms CNNs. However, their performance deteriorates when objects are occluded. Apart from the possibility of losing some key points, occlusion leads to a huge loss of context and semantics. Yet the context and semantics are critical for accurate 2D HPE.

In this paper, we propose an occlusion augmentation method to artificially hide random parts of the human body in the image. Training with random occlusion, the models present strong generalization ability when encountering occlusions in the wild. To verify our augmentation method, we reproduce three well-known baselines, i.e. the SimpleBaseline [18], HRNet [16], and the ViT-Pose [19]. In the experiments, we first show that their performances drop when test images are occluded to support our motivation. Then the boost of accuracy is presented when models are fine-tuned with our occlusion augmentation method. Furthermore, detailed analysis and insights are shared.

2. Methods

In this section, we would first introduce our artificial occlusion transformation. Then a brief introduction to PoseResnet, HRNet, and ViTPose is presented.

2.1. Artifical Occlusion Transformation

To imitate in-the-wild occlusions, several augmentations of occlusions can be generated and applied to images. We can apply occlusions with different positions, sizes, shapes, and colors. The colors in the corresponding occluded area are replaced with a randomly selected color. Intuitive examples are provided in Fig. 1.

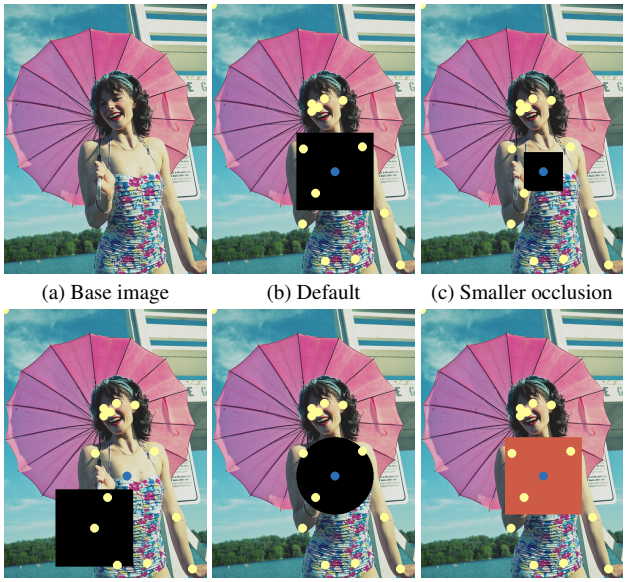
We introduce two variables, $p_{occ} \in [0, 1]$ and $\alpha \in [0, 1]$, to control the probability of a sample being occluded and the size of the occlusion, respectively. p_{occ} can be interpreted as the dataset distribution, wherein $p_{occ} = 0.75$ means that 75% of the dataset is occluded. We randomly initialize coordinates for the occlusion shape based on the bounding box coordinates of the subject's skeleton. α controls the offset distance between the shape and skeleton coordinates, and with respect to the length and width of the aforementioned bounding box. A larger α value constricts the shape horizontally and vertically, i.e., a smaller area will be occluded. Thus, varying α changes the size of the occlusion while ensuring that it covers the subject of the input image. Default training settings are kept as $p_{occ} = 0.5$ and $\alpha = 0.1$. During testing, $p_{occ} = 1.0$ and $\alpha = 0.1$.

In the experiment section, we will first check how occlusions would degrade the performance of the reproduced models. Then, we will explore how to boost the robustness of those models by fine-tuning models with the proposed occlusion transformation.

2.2. Simple Baselines: PoseResnet

Compared to previous methods [5, 13], the Simple Baselines [18] model, PoseResnet, is much simpler architecturally while achieving state-of-the-art performance in pose estimation. Specifically, the network is composed of a ResNet backbone with three additional deconvolutional layers (with batch normalization and ReLU activation) over the final convolution stage. Each layer consists of 256 filters, a kernel of size 4×4, and the stride is set to 2. The simplicity of the architecture is mainly due to the inclusion of deconvolutional layers — it combines the feature map upsampling and convolutional parameters into the same layers, as well as removes the usage of skip layer connections.

To generate predicted heatmaps $\{H_1, ..., H_k\}$, where k represents the number of key points, a final 1×1 convolutional layer is added to the end of the model. Targeted heatmaps, on the other hand, are generated by applying a 2D gaussian that is centered on the ground truth location of each joint. The loss function used is Mean Squared Error (MSE) to compute the difference between predicted heatmaps and targeted heatmaps.



(a) Base image (b) Default (c) Smaller occlusion

(d) Right elbow center (e) Ellipse shape (f) Red colour

Figure 1. Occlusion transformation applied on one image example

2.3. HRNet

HRNet [14], or High-Resolution Net, aims to maintain high-resolution representations by connecting convolutions of high-to-low resolutions in parallel. Each resolution subnetwork (i.e., stage) consists of a sequence of convolutions and a down-sample layer. While existing networks follow a *sequential* fashion for their subnetworks, HRNet employs *parallel* multi-resolution subnetworks. Starting from a high-resolution subnetwork, high-to-low-resolution subnetworks are added to form new stages. Each multi-resolution subnetwork is then connected in parallel to form a pyramid-like structure, where later stages inherit the resolutions of the former stage, as well as an additional lower resolution. *Repeated multi-scale fusion* then occurs, in which parallel subnetworks repeatedly exchange information with other subnetworks through multi-resolution group convolutions. This exchanging of information is enabled by *exchange units*. Repeated multi-resolution fusions yield high-resolution representations that are richer in information as they are boosted by low-resolution representations.

HRNet is composed of 4 parallel subnetworks and comes in two sizes, HRNet-W32 and HRNet-W48. The numeral in the name denotes the width of the first of the four stages. There are 8 exchange units in the network.

2.4. ViTPose

The success of vision transformers [7, 11, 21] has motivated their applications for the pose estimation task, such

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Mean
PoseResnet	96.4	95.3	89.0	83.1	88.4	84.0	79.6	88.5
	84.5	72.1	64.0	56.2	64.9	61.9	63.5	67.2
HRNet	97.1	95.9	90.3	86.4	89.1	87.1	83.3	90.3
	83.6	66.2	57.4	49.6	57.2	56.6	60.1	61.9
ViTPose	97.6	97.4	93.7	90.1	92.4	91.9	88.3	93.4
	97.0	95.5	89.2	82.6	88.0	87.8	86.1	89.7

Table 1. Performance comparison of pretrained models. PCKh@0.5 scores for the non-occluded test set (first row) and for our proposed occluded test set (second row) are reported.

as TokenPose [9] and TransPose [20]. Most transformers adopt a CNN as a backbone for extracting features and a transformer of elaborate structures for refining these features and modeling the relationship between key points of the body. Contrary to these sophisticated designs, ViTPose [19] revealed that plain vision transformers could achieve superior performance meanwhile having the advantages of simplicity, scalability, flexibility, and transferability. The image is first embedded by 16×16 patches in 768 channels and then processed by one encoder and one decoder. The encoder consists of 12 stacked transformer blocks, each of which is formed by a multi-head self-attention (MHSA) layer, a feed-forward network (FFN), and layer normalization layers. Two kinds of lightweight and effective decoders were proposed by [19]: the classic decoder is composed of two deconvolution blocks, each block upsamples the features maps by 2 times; the simple decoder directly upsamples by 4 times with bilinear interpolation. The localization heatmaps for the key points are obtained by a convolution layer (kernel size 1×1 for the classic decoder, 3×3 for the simple decoder) after the upsampling.

3. Experiments

In this section, we first briefly introduce the experiment setting. Then the models performances with and without occlusions are presented. Later, we show how fine-tuning with the occlusions transformations can help boost models performances. Besides, detail analysis is provided.

3.1. Experiment Settings

Evaluation Metric. Among a variety of evaluation metrics for 2D HPE summarized in [23], we have focused on the following metrics:

- Percentage of Correct Keypoints (PCK). PCK measures the accuracy of the predicted key point and the ground truth joint within a certain distance threshold. Typical thresholds used are PCKh@0.5 (50% of the head bone link), PCK@0.2 (20% of the torso diameter), and 150 mm.
- Average Precision (AP) and Average Recall (AR). Based on the aforementioned key point detection results, precision (the ratio of true positive results to the total positive results) and recall (the ratio of true positive results to the total number of ground truth positives) can be obtained. AP computes the average precision value for recall over 0 to 1. The COCO evaluation metrics also report AP across scales: small (AP^S), medium (AP^M), and large (AP^L). Respectively, each scale represents the AP score of objects that cover below 32^2, between 32^2 and 96^2, and above 96^2 areas, where the area is measured as the number of pixels in the image segmentation mask.

Datasets. We choose the MPII dataset [2] and the COCO dataset [10] for experiments. The MPII human pose estimation dataset consists of 28k training and 11k test images, covering human activity under different viewing angles. The common objects in context (COCO) dataset contains 330k images in total, while 200k of them are labeled.

In addition to the pure image dataset, we need ground-truth labels for the human pose joints. The *bearpaw/pytorch-pose* [1] GitHub repository provides those labels for COCO and MPII, converted to python-friendly JSON files. Those labels consist of the corresponding image name, the image width, and height, the body position within the image, the relative body scale, as well as all joint positions sorted as { nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle } for COCO (similar for MPII).

Data Augmentations. Apart from occlusions, we also augment the data during training with random rotation ($[-45°, 45°]$), random scale ($[0.65, 1.35]$), and flipping following contemporary work [14, 18, 19]. We retain similar parameters for training and testing as the original studies for fair comparison in model performances. For PoseResnet and HRNet, ResNet-50 is used as the default backbone. HRNet-W32 and ViTPose-B are used by default and simply referred to as HRNet and ViTPose for the rest of the paper.

3.2. Pose Estimation on MPII

We evaluate the performance of three models, PoseResnet, HRNet, and ViTPose, on the MPII in various occlusion settings. First, it is important to note the significant drop in performance when testing the performance of models trained on non-occluded samples (i.e., the original train and test set) on our occluded test set. As Table 1 shows, the drop in performance ranges from 4% to 30% in different models. It is interesting to note that while HRNet boasts superior performance under non-occluded settings as compared to PoseResnet, the drop in performance of the former is much sharper when occlusion is introduced. Amongst the three models, ViTPose is most robust towards occlusion as suggested by the least drop in performance. This is probably due to MAE's ability to reduce the noise of input data.

Method	p_{occ}	Mean (O)	Mean (NO)
PoseResnet	0	65.5	86.2
	0.25	80.4	86.2
	0.5	80.8	86.3
	0.75	**81.2**	**86.7**
	1.0	**81.2**	85.6
HRNet	0	66.1	87.2
	0.25	83.0	88.1
	0.5	83.5	88.1
	0.75	83.8	**88.1**
	1.0	**84.3**	87.4
ViTPose	0	81.5	**89.3**
	0.25	82.4	88.9
	0.5	**82.8**	**89.3**
	0.75	82.7	89.1
	1.0	82.4	88.9

Table 2. Performance comparison of models trained with different p_{occ} values, then tested on the occluded test set (O; $p_{occ} = 1$) and non-occluded test set (NO; $p_{occ} = 0$). Evaluation metric used is PCKh@0.5.

By masking random patches of the image, which is similar to our occlusion settings, the model is able to reconstruct the missing patches and learn a better representation of the input images.

Table 2 reports the pose estimation performance of each model under various occlusion settings. The NO column represents the ideal scenario where the entire human body is visible in an image, whereas the O column resembles real-life settings where parts of the human body might be blocked by other human or obstacles. Under the occluded test set setting (O column), it is observed that model trained with non-zero p_{occ} has a higher score than model trained with zero p_{occ}, which supports our initial hypothesis - a model trained for a difficult task is performing better as compared to a model trained for a easy task. While all three models perform better using occluded training set, the performance gap for ViTPose is relatively smaller than those in PoseResnet and HRNet, which suggests that transformer based methods are inherently more robust than CNN based methods, therefore less occlusion is needed during training. Under the non-occluded test set setting (NO column), it appears that there is minimal to no benefit when using higher p_{occ} values, *i.e.*, more occluded samples in train set. The overall results imply that the choice of p_{occ} value for optimal performance depends on the actual data distribution (% of occluded data in the test set) and the type of model used (CNN based or transformer based).

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
PoseResnet	72.4	91.5	80.4	69.7	76.5	75.6
	45.6	69.1	48.9	46.4	45.1	48.5
HRNet	76.5	93.5	83.7	73.9	80.8	79.3
	44.2	66.0	47.8	45.6	43.1	47.1
ViTPose	75.8	90.7	83.2	72.3	82.6	81.1
	69.8	87.7	77.1	65.3	77.4	75.2

Table 3. Performance comparison of pretrained models on the COCO test set, without occlusion (first row) and with occlusion (second row). Occlusion settings follow MPII implementation.

3.3. Pose Estimation on COCO

We extended our experiments to the COCO dataset and observed a similar decline in model performance due to occlusion in the test set across all metrics and all models. Table 3 shows that the drop in model performance for HRNet is rather significant when occlusion is introduced. The effect on ViTPose is the smallest. Due to computational limits as size of COCO is considerably larger than MPII, we leave further investigation on optimal distribution for future work.

4. Conclusion & Future Directions

We present a new dataset to evaluate the performance of 2D human pose estimation methods under various occlusion settings and show the importance of achieving robustness in this aspect. Our results show that while existing state-of-the-art methods excel in the original problem task, they are unable to perform well when the human subject is partially occluded.

In this study, we fix $\alpha = 0.1$ and examine the performance of different occlusion settings on MPII data. We leave further investigation of optimal values and robustness under stronger occlusion settings on COCO for future work. Extending the problem set to 3D human pose estimation can also be an interesting direction, as viewing a subject from different angles may alleviate occlusion. Another possible extension is to evaluate performance in multi-person pose estimation with occlusion. Finally, it would be beneficial to build occlusion datasets for non-human pose estimation as well.

References

[1] https://github.com/bearpaw/pytorch-pose. [Online; accessed 10.11.2022]. 3

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 3

[3] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5933–5942, 2019. 1

[4] Haoming Chen, Runyang Feng, Sifan Wu, Hao Xu, Fengcheng Zhou, and Zhenguang Liu. 2d human pose estimation: A survey, 2022. 1

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. 2018. 2

[6] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[8] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 1

[9] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11313–11322, 2021. 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 3

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[12] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001. 1

[13] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision – ECCV 2016*, pages 483–499. Springer International Publishing, 2016. 2

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

[15] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 1

[16] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 1

[17] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11088–11096, 2020. 1

[18] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3

[19] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022. 1, 3

[20] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021. 3

[21] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3417–3425, 2022. 2

[22] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *CoRR*, abs/2012.13392, 2020. 1

[23] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *CoRR*, abs/2012.13392, 2020. 3