Robust Data Augmentation for Accurate Human Pose Estimator

Tien-Dat Tran, Xuan-Thuy Vo, Adri Priadana, and Kang-Hyun Jo*

School of Electrical Engineering, University of Ulsan, Ulsan 44610, South Korea
{tdat,xthuy}@islab.ulsan.ac.kr; priadana3202@mail.ulsan.ac.kr:
acejo@ulsan.ac.kr

Abstract. Accurate occluded key point identification is a challenge and hot topic for human pose estimation. To make the occluded or invisible keypoint better, data augmentation play an important role which makes the network overcome complex case. In this paper, we want to apply cut-out technique which is a strong method to tackle the problem. Furthermore, data augmentation demonstrates its superiority over other methods without enlarging the computational cost. Correspondingly, the proposed work focuses on powerful data augmentation for occluded keypoints. First, following a human detection in the detector network, feed the human proposal region into the data augmentation, which makes the network can learn more about the occluded cases. The data after data augmentation then apply to train for the pose estimator. The estimator collects more information in occluded keypoints, illustrating higher precision efficiency. The outputs of our experiments would also demonstrate a distinction between the use of cut-out data augmentation and existing approaches. The predicted joint heatmaps are more accurate than the baseline technique despite using the same amount of parameters due to the transition to a high-resolution network (HRNet) for the pose estimator. In terms of AP, the suggested design outperforms the baseline-HRNet by 0.2 points, but in the occluded case, the pose estimator performs much more better. Additionally, the COCO 2017 benchmarks, which are now accessible as an open dataset, were used to train the proposed network.

Keywords: Deep Learning · Occlusion Keypoint · Data Augmentation · Human pose estimation.

1 Introduction

In the modern world, 2D human pose estimation plays a crucial role but challenging function in computer vision, which can serve numerous objectives such as human robotics [23, 3], activity recognition [6, 8], human re-identification [25, 11], or film industry[2, 10]. Human pose's main goal is to identify bodily sections for human body joints.

* Corresponding author

Fig. 1. Occlusion Keypoint in the testing on the MPII dataset. The red dot is the occluded keypoint which is one of big challenges nowsaday for human pose estimation

In human pose estimation, there are many challenges that attack the network performance. Among the challenges, the occluded keypoint shown in Fig.1 is one of the biggest challenges for the network training to get better performance. To solve this kind of problem many researchers used another network such as a graph neural network[17] or Generative adversarial network[21] to generate a new structure for the human pose to train. However, utilizing a new network for the occluded problem is costly. To solve the problem, data augmentation is a potential candidate that can remedy the challenges, which is not consumed much more resources than using another network. Data augmentation does not only enhance the value of information from the image but also not consume more parameters in the training process. In more detail, the data augmentation performs a global transformation for the images. The transformation makes the network get many extra points of view about images, which show a lot of improvement[4]. Besides all of the advantages, data augmentation also brings extra unimportant data, which makes the data redundant. On the other hand, many kinds of data augmentation such as crop makes the data much more margin or rotate can make the data lost information. Hence, choosing the suitable for data augmentation is really important to make the network can get better performance.

In the proposed work, we make a deep investigation into data augmentation which compares the original method and a new one. The original data augmentation[19] apply flip, rotate, scale, and half body transform. This kind of method can enhance the accuracy of keypoint however for the occluded keypoint, it shows their disadvantages which can not significantly improve the accuracy of the occluded keypoint. Hence, the proposed research applies a new kind of data augmentation which call cut-out. By using cut-out method, the whole architecture can gain more accuracy, especially in the case of occluded keypoint which can check at the experiment result.

In particular, the proposed study was based on a simple framework [4], which applies the top-down method for human pose estimation. Without taking the data much more different from the original but more occlude cases appear, the proposed network can be easy to learn the invisible keypoint. For instance, with the extra training data, the network may learn to connect the keypoint for the visible part such as occlude wrist or ankle keypoint. Furthermore, the number

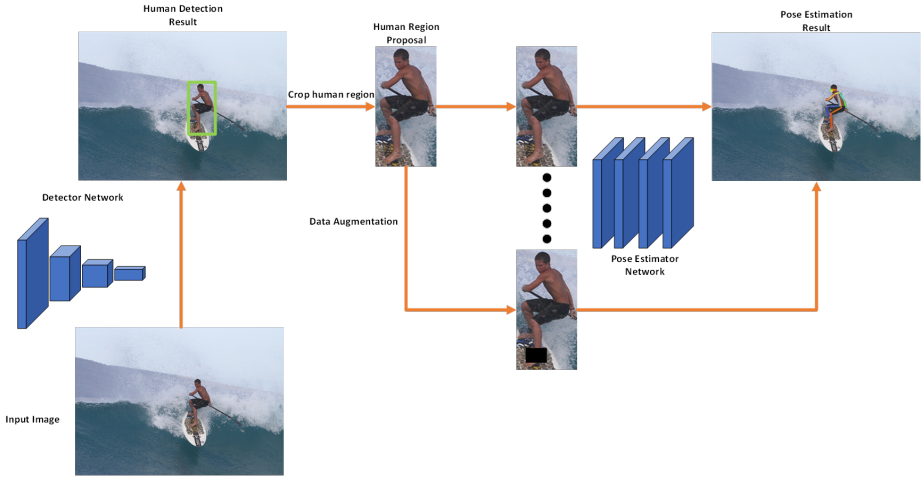

Fig. 2. Full system of 2D human pose estimation from input to pose estimation. The proposed approach split the system into 2 stages, the first stage is the human detector and the second stage is the pose estimator

of parameters was not changed, resulting in network speed not changing while the accuracy for the occluded keypoint improved much more.

To make clear about the cut-out augmentation, the transformation can apply for all of the pose estimators apply the data augmentation. Also, this method is easy to apply not only for estimator but also detector

In summary, the main contribution of the paper describes in two-fold:

• We design and apply a new data augmentation called the cut-out that makes the data more information about the occluded problem.

• We comprehensively evaluate and compare the proposed method with the original method on the COCO benchmark dataset, which is the most popular dataset for keypoint.

2 Related work

2D-Human Pose Estimation The most important aspect of human pose estimate is joint detection and its interaction with spatial space, as seen in Fig.3. There are two main methods applied for human pose estimation, which is the bottom-up and top-down method. For the bottom-up method, Deeppose[22], Simple baseline makes use of joint prediction using an end-to-end network with a larger parameter. Later, Newell with the Stacked hourglass network [13] reduces the number of settings while maintaining great accuracy. To represent local joints, all of the approaches employed Gaussian distributions. After that, a convolution neural network was utilized to estimate human posture estimation. For the top-down method, first, we apply a detector for the human proposal region, and after that using the crop region for pose estimation. Because top-down

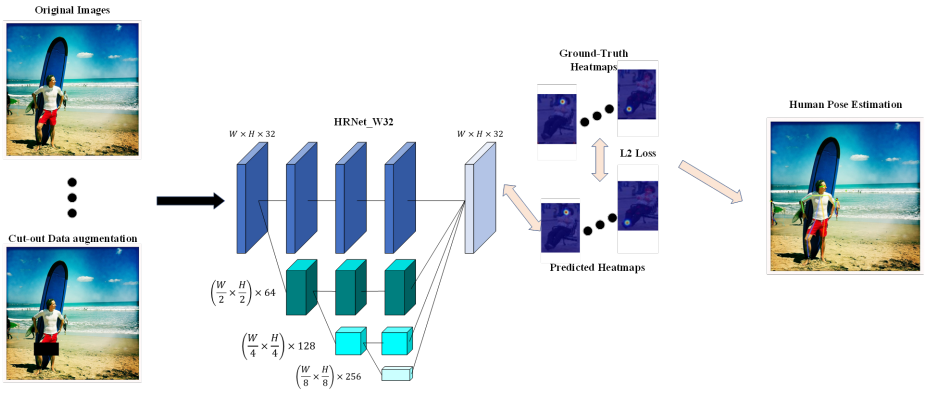

Fig. 3. Illustrating the architecture of the proposed 2D-human-pose estimator. The proposed network training with the original images and transformation images

method is using the detector so the accuracy can get better than the bottom-up. And bottom-up is an end to end method so the inference time can be better than the top-down.

In the proposed paper, we apply the top-down method for the whole architecture which illustrates in the Fig.2, From the input images, the model utilizes the existing detector for human detection. YOLO[20] is one of diversity kind detector, which has many versions for different cases such as real-time, high accuracy, or for mobile devices. To balance everything, the proposed method utilizes the YOLO-V3. After applying the detector to the human region, the whole network utilizes the pose estimator to perform training tasks in the human region. Additionally, data augmentation will apply in this stage. In comparison, the top-down method uses sufficient perspective for network design, with a limited number of parameters and high speed or a larger number of parameters and lower speed.
Data Augmentation: Data augmentation play an important role in computer vision task which compensate for the lack of data in real life. From the original input image, data augmentation makes more data for the network can learn from many perspective views. Notice that the more diversity in the dataset, the more accuracy for human detection. Moreover, data augmentation did not increase the number of parameters so the computational cost will not increase. However, data augmentation also can drive the detector worth[16] which make the detector hard to learn the feature of images, especially for occluded keypoint.

Most detector networks used the same data augmentation such as Flip, Rotation, Scale zoom in and zoom out or half body transforms with a probability of 0.3. However, this kind of data augmentation does not work well with occluded keypoint. Hence, we apply the cut-out augmentation to show the real case to build the network can learn more about the occluded keypoint. Furthermore, the occluded and invisible keypoint appear more in the data so that the

network learns better. To improve accuracy, the cut-out method shows better performance in the data augmentation tasks.

3 Methodology

3.1 Network architecture

Detector The human detector plays an important role in the whole system. First, input image matrix $\alpha(\mathbf{X})$ feed to the human detector. After that, the detector gives the result of the human region $\beta(\mathbf{X}')$ which is the subset of $\alpha(\mathbf{X})$.

$$D\{\alpha(\mathbf{X})\} = \beta(\mathbf{X}') \tag{1}$$

Following resize function make $\beta(\mathbf{X}')$ into 256×192 images which can call $\gamma(\mathbf{X}')$

$$\gamma(\mathbf{X}') = Resize(256 \times 192, \beta(\mathbf{X}')) \tag{2}$$

The proposed study utilizes YOLO-V3[18] for the main detector in the whole architecture. The YOLO-V3 is the medium detector that can balance the computational cost and accuracy.

Data Augmentation In the proposed paper, we apply one more data augmentation for the bounding box $\gamma(\mathbf{X}')$ after the detector stage. Besides the original data augmentation which includes Flip, Rotate, Scale, and Half Body Transform, additional cut-out augmentation is applied. First, the cut-out function can be understood

$$C\{\gamma(\mathbf{X}')\} = Cutout(\gamma(\mathbf{X}'), n, p) \tag{3}$$

with p is the padding fraction for cut out, which is the number of pixels applied for cut out. n is the number of cutout pads in the human region. In the training process, we set n equal to 1. For more detail, the padding p is set random base on the size of $\gamma(\mathbf{X}')$ which is 256×192. The human region $\gamma(\mathbf{X}')$ have the coordinate of x_{min}, y_{min} and x_{max}, y_{max} which is the coordinate of the human region in the images. the padding P will take random with the condition $x_{min} \leq P_x \leq x_{max}$ and $y_{min} \leq P_y \leq y_{max}$.This research applies the Clamp function in Pytorch[15] to make the border for the cut-out pad inside the human region $\gamma(\mathbf{X}')$. After having the pad for cut-out we use the replace function to apply the pad to the human region. Finally, the cut-out image and the original pad will apply for the training part in the pose estimator

Pose estimator The pose estimator use backbone mainly HRNet-W32 and HRNet-W48 [19]. Fig.3 shows our proposed architecture for the estimator which is based on the backbone. The estimator HRNet includes 4 stages, which consist of residual blocks and connections. The input is the human region proposal from the detector resize the size to 256×192 for both HRNet. After that, each residual block is traversed by the feature maps, and each stage's $W \times H$ resolution is reduced twice. The size of the output tensor is finally reduced to $\frac{W}{16} \times \frac{H}{16}$ with 256 channels at the last bottom layer of the network after traveling down the spine. The first subnetwork, whose size is $W \times H$, is the only one that the backbone

network will employ during the regression. Additionally, each stage would see a doubling of the channel size. After the first block, it increases from 32 to 256 in the last layer. In order for the Training System to predict the human joints, the backbone network must gather data and feature maps from the input image by utilizing the cross entropy loss which describes in the Loss function part.
After extracting the information using the backbone network, the upsampling network recovers the information by taking the feature map from the final layer of the backbone network and upsampling it. Following that, the feature map will be trained with Ground-truth Heat Maps, as shown in Fig.4. The default heat map size is a quarter with the original images 256×192 for HRNet-W32 and 384×288 for HRNet-W48. However, we resize the input image for HRNet-W48 into 256×192 to save the parameter and time for training. For regression, the proposed study will use these heat maps and the ground truth heatmap to create the predicted keypoint. This article employs HRNet so the feature maps are kept to the shape with the original input(in Figure 3). The residual block contains both batch normalization and ReLU[7].

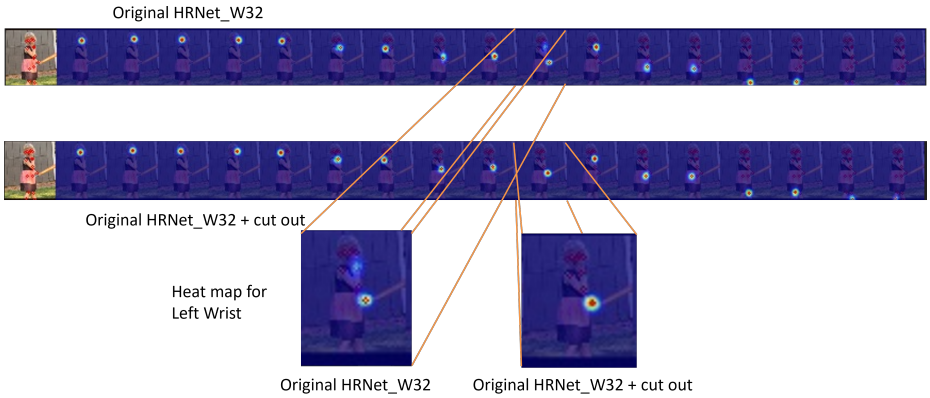

Fig. 4. Heat-map generates in pose estimator before and after applying the cut-out data augmentation. In comparison, the visible keypoint is almost the same in both cases. The occluded keypoint are much more different can show for the left wrist

3.2 Loss Function

Heat maps are used in this work to illustrate body joint locations for the loss function. As the ground-truth position in Fig. 4 by $a = \{a_k\}\, k = 1^K$, where $a_k = (x_k, y_k)$ is the spatial coordinate of the kth body joint in the image. The ground-truth heat map value H_k is then constructed using the Gaussian

distribution with the mean a_k and variance $\sum$ as shown below.

$$H_k(p) \sim N(a_k, \sum) \tag{4}$$

where $\mathbf{p} \in \mathbf{R^2}$ represents the coordinate, and $\sum$ is experimentally defined as an identity matrix $\mathbf{I}$. The last layer of the neural network predicts K heat maps, *i.e.,* $\hat{S} = \left\{\hat{S}_k\right\} k = 1^K$ for K body joints. A loss function is defined by the mean square error, which is calculated as follows:

$$L = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \left\| S_k - \hat{S}_k \right\|^2 \tag{5}$$

N denotes the number of samples in the training session. Using information from the backbone network's last layer, the network generated prediction heat maps using ground-truth heat maps.

4 Experiments

4.1 Experiment Setup

Dataset. The Microsoft COCO 2017 dataset [14] is used throughout the studies in the proposed network. The data collection contains 250K human samples and 200K images, each human identity has 17 keypoint labels. Three folders, labeled train set, validation set, and test-dev set, contain training, validation, and testing photographs respectively. In addition, the original is available to view, and the validation and training annotations are as well.
This study also made use of a commercial dataset that records footage of individuals working in a commercial laboratory setting. The dataset consists of 4 films with frame rates ranging from 4000 to 6000. There are several difficulties in the video, including overlapped people, crowded at scenes, and little people. Therefore, it is possible to test how effective the suggested strategy is at tracking.
Evaluation metrics. For COCO[12], the proposed study used Object Keypoint Similarity (OKS) using $OKS = \frac{\sum_i exp(-{d_i}^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$. Specifically, d_i represents the Euclidean separation between the predicted keypoint and the ground truth, v_i represents the target's visibility flag, s represents the object scale, and k_i represents a keypoint for each joint. Next, the average accuracy and recall scores are calculated. Table I shows the AP and AR averages from OKS=0.5 to OKS=0.95, with AP^M standing for medium objects and AP^L for large objects.
Implementation details All experiments are carried out using the codebase called AlphaPose[4] and tested on two datasets. The picture input resolution was reduced to 256×192. The model was trained using CUDA 10.2 and CuDNN 7.3 on a single NVIDIA GTX 1080Ti GPU.

The method included data augmentation in model training, including flip, rotation at 40 degrees by design, and scale with the factor was set at 0.3. Set

the batch size to 4 and use the shuffle function when using training photos. In our experiment, there are 210 total epochs, and the base learning rate is set at 0.001 before being multiplied by 0.1 (learning rate factor) at the 170^{th} and 200^{th} epochs. The Adam optimizer, [9], was used, and the momentum is 0.9.

4.2 Experiment Result

COCO datasets result The proposed method compares each circumstance while adding different kinds of data augmentation for the pose estimator, as shown in Table 1. The Average Precision (AP) demonstrates that using the proposed method for cut-out gains 0.6 in mAP, which boosts accuracy by 1 percent. Furthermore, this study also investigates again another data augmentation[12], which is set up in almost a training process for pose estimator. The default data augmentation including Flip, Rotation, Scale, and Half body transform is trained again separately and shown in Tab.1. In total, when combining all of the data augmentations and apply the cut-out the AP slightly increase.

Table 1. The result for applying different kinds of data augmentation in HRNet

Backbone	Data augmentation	mAP
HRNet-W32	Without	72.9
HRNet-W32	Flip	73.6
HRNet-W32	Rotate	73.4
HRNet-W32	Scale	73.3
HRNet-W32	Half body transform	73.7
HRNet-W32	Cut-out (our)	73.5
HRNet-W32	All	74.6

Table 2. Comparison on COCO Validation Dataset

Method	Backbone	Input size	#Params	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-Stage Hourglass[13]	8-Stage Hourglass	256×192	25.1M	66.9	-	-	-	-	-
Mask-RCNN[5]	ResNet-50-FPN	256×192	-	63.1	87.3	68.7	57.8	71.4	-
OpenPose[1]	-	-	-	61.8	84.9	67.5	57.1	68.2	66.5
PersonLab[14]	-	-	-	78.7	89.0	75.4	64.1	75.5	75.4
SimpleBaseline[24]	ResNet-50	256×192	34.0M	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline[24]	ResNet-101	256×192	53.0M	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline[24]	ResNet-152	256×192	68.6M	73.7	91.9	81.1	70.3	80.0	79.0
HRNetBaseline[19]	HRNet-W32	256×192	28.5M	74.4	90.5	81.9	70.8	81.0	79.8
HRNetBaseline[19]	HRNet-W48	256×192	63.6M	75.1	90.6	82.2	71.5	81.8	80.4
HRNet + our cut-out	HRNet-W32	256×192	28.5M	74.6	90.7	82.1	71.1	81.2	80.0
HRNet + our cut-out	HRNet-W48	256×192	63.6M	75.3	90.7	82.4	71.8	81.9	80.7

In Table 2, the proposed result was estimated on the COCO validation dataset. The AP in the suggested approach is greater than the Basic High-Resolution benchmark in all situations of 0.2 AP in both backbone HRNet-32 and HRNet-W48. Furthermore, the average recall (AR) is 0.3 points higher in the case of HRNet-W32 and 0.2 points higher in the case of HRNet-W48. In total, the experiment results slightly increased in both AP and AR but it significantly improve in the case of occluded keypoint. The visual result for heatmap detail can see in Fig.4 which shows that applying the proposed research makes the predicted heat map get more accurate. Fig. 5 shows the qualitative result for the COCO 2017 dataset with 2 same images as the original pose estimator and proposed technique. For more detail, the green box means more occluded keypoint got detection while the red box means the wrong keypoint predict
Industrial datasets result: The proposed research will focus on the industrial environment in future work. Hence testing the pose estimation with diversity environment is necessary. This study tests on the industrial dataset that contains 200 images for the occluded challenge. The result shown in Fig.6 for the original and the improvement after apply the data augmentation.

5 Conclusion

This research shows the effect of the data augmentation on CNNs especially for occluded human keypoint, with a focus on cut-out for human proposals. Furthermore, our work demonstrates that not increasing the computation cost, the data augmentation utilized has a more considerable effect. Moreover, the cut-out focused more on the essential feature map than the other element. The network will become more effective as a consequence, particularly for various computer vision-related tasks.

Along with many other modern designs, human pose estimation has a number of problems that need to be solved. The pictures' concealed joints, which were challenging to train for and predict, were the first problem. Second, joints in the human body must be accurately eliminated from low-resolution human images. The pictures that follow show crowd situations, when it is usually challenging to pinpoint where each participant's joints are located. Last but not least, there is a lack of data on photos with missing pieces for assessing human postures. The proposed method tries to solve the first problem which is also the hardest case compared to all of the problems. Hence, future research will try to focus on the remained problem. Moreover, find a way to apply the method in the surveillance system.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government. (MSIT)(No.2020R1A2C2008972)

Fig. 5. Qualitative images for human pose estimation in COCO2017 test-dev set. In two similar images, the Right side is the result of the original human pose estimator HRNet-W32, and Left side is our approach. Greenbox means better keypoint detection than normal. Redbox means inaccurate keypoint detection.

Fig. 6. Testing on the industrial dataset

References

1. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields (2016). https://doi.org/10.48550/ARXIV.1611.08050, https://arxiv.org/abs/1611.08050
2. Chen, C., Ramanan, D.: 3d human pose estimation = 2d pose estimation + matching. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5759–5767 (July 2017). https://doi.org/10.1109/CVPR.2017.610
3. Chou, C.J., Chien, J.T., Chen, H.T.: Self adversarial training for human pose estimation (2017)
4. Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time (2022). https://doi.org/10.48550/ARXIV.2211.03375, https://arxiv.org/abs/2211.03375
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn (2017)
6. Hussain, Z., Sheng, M., Zhang, W.E.: Different approaches for human activity recognition: A survey (2019)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
8. Kim, E., Helal, S., Cook, D.: Human activity recognition and pattern discovery. IEEE Pervasive Computing **9**(1), 48–53 (Jan 2010). https://doi.org/10.1109/MPRV.2010.7
9. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
10. Li, S., Ke, L., Pratama, K., Tai, Y., Tang, C., Cheng, K.: Cascaded deep monocular 3d human pose estimation with evolutionary training data. CoRR **abs/2006.07778** (2020), https://arxiv.org/abs/2006.07778
11. Li, W., Zhao, R., Wang, X.: Human reidentification with transferred metric learning. In: Asian Conference on Computer Vision (ACCV). pp. 31–44 (11 2012)
12. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014), http://arxiv.org/abs/1405.0312
13. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR **abs/1603.06937** (2016), http://arxiv.org/abs/1603.06937
14. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model (2018). https://doi.org/10.48550/ARXIV.1803.08225, https://arxiv.org/abs/1803.08225
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
16. Pytel, R., Kayhan, O.S., van Gemert, J.C.: Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions (2020). https://doi.org/10.48550/ARXIV.2010.10451, https://arxiv.org/abs/2010.10451
17. Reddy, N.D., Vo, M., Narasimhan, S.G.: Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7326–7335 (2019)

18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018). https://doi.org/10.48550/ARXIV.1804.02767, https://arxiv.org/abs/1804.02767
19. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation (2019)
20. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. CoRR **abs/1911.09070** (2019), http://arxiv.org/abs/1911.09070
21. Tian, L., Liang, G., Wang, P., Shen, C.: An adversarial human pose estimation network injected with graph structure (2021). https://doi.org/10.48550/ARXIV.2103.15534, https://arxiv.org/abs/2103.15534
22. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. CoRR **abs/1312.4659** (2013), http://arxiv.org/abs/1312.4659
23. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines (2016)
24. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. CoRR **abs/1804.06208** (2018), http://arxiv.org/abs/1804.06208
25. Yang, X., Wang, M., Tao, D.: Person re-identification with metric learning using privileged information. CoRR **abs/1904.05005** (2019), http://arxiv.org/abs/1904.05005