

HPRNet: Hierarchical Point Regression for Whole-Body Human Pose Estimation

Nermin Samet, Emre Akbas
Department of Computer Engineering
METU
{nermin,emre}@ceng.metu.edu.tr

Abstract

In this paper, we present a new bottom-up one-stage method for whole-body pose estimation, which we call “hierarchical point regression,” or HPRNet for short. In standard body pose estimation, the locations of ~ 17 major joints on the human body are estimated. Differently, in whole-body pose estimation, the locations of fine-grained keypoints (68 on face, 21 on each hand and 3 on each foot) are estimated as well, which creates a scale variance problem that needs to be addressed. To handle the scale variance among different body parts, we build a hierarchical point representation of body parts and jointly regress them. The relative locations of fine-grained keypoints in each part (e.g. face) are regressed in reference to the center of that part, whose location itself is estimated relative to the person center. In addition, unlike the existing two-stage methods, our method predicts whole-body pose in a constant time independent of the number of people in an image. On the COCO WholeBody dataset, HPRNet significantly outperforms all previous bottom-up methods on the keypoint detection of all whole-body parts (i.e. body, foot, face and hand); it also achieves state-of-the-art results on face (75.4 AP) and hand (50.4 AP) keypoint detection. Code and models are available at <https://github.com/nerminsamet/HPRNet>.

1. Introduction

As a challenging computer vision task, human pose estimation aims to localize human body keypoints in images and videos. Human pose estimation has an important role in several vision tasks and applications such as action recognition [12, 32, 56, 18, 35], human mesh recovery [9, 30, 22, 26], augmented/virtual reality [10, 14, 55], animation and gaming [2, 1, 3, 29]. Unlike the standard human pose estimation task, whole-body pose estimation aims to detect face, hand and foot keypoints in addition to the

standard human body keypoints. The challenge in this problem is the extreme scale variance or imbalance among different whole-body parts. For example, the relatively small scale of face and hand keypoints make accurate localization of face and hand keypoints more difficult compared to the standard body keypoints such as elbow, knee and hip. Direct application of existing human pose estimation methods do not yield satisfactory results due to this scale variance problem.

Even though human pose estimation has been well studied for the past few decades, the whole-body pose estimation task has not been sufficiently explored, mainly due to the lack of large-scale fully annotated whole-body keypoint datasets. The previous few methods [6, 17], trained several deep networks separately on different face, hand and body datasets, and ensembled them during inference. These methods suffer from issues arising from datasets’ biases, variations of illumination, pose and scales, and complex training and inference pipelines.

Recently, in order to address the missing benchmark issue, Jin *et al.* [25] introduced a novel dataset for whole-body pose estimation, called COCO WholeBody. COCO WholeBody extends COCO keypoints dataset [34] by further annotating face, hand and foot keypoints. In addition to the standard, 17 human body keypoints from the COCO keypoints dataset; 68 facial landmarks, 42 hand keypoints and 6 foot keypoints are annotated (Figure 1). Along with these 133 whole-body keypoint annotations, the dataset also has face and hand bounding box annotations that were automatically computed from the extreme keypoints of the corresponding part. They also proposed a strong baseline, called ZoomNet, which has set the state of the art. ZoomNet is a top-down, two-stage method based on the human pose estimation model HRNet [50]. Given an image, ZoomNet first detects person instances using the FasterRCNN [45] person detector, then it predicts 17 body and 6 foot keypoints using a CNN model. Later, to overcome the scale variance between whole-body parts, ZoomNet crops the hand and face areas that it detected and transforms them to higher resolu-

tions using separate CNNs to further perform face and hand keypoint estimation.

There are two main approaches for human pose and whole-body pose estimation; *bottom-up* [7, 38, 36, 37, 28, 41, 4, 44, 17, 19, 20, 21, 23, 24] and *top-down* [8, 42, 16, 15, 50, 54]. Bottom-up methods directly detect human body keypoints and later group them to obtain final poses for each person in a given image. On the other hand, top-down methods (e.g. ZoomNet) first detect and extract person instances, then apply pose estimation on each instance separately. The grouping stage of bottom-up methods is more efficient than repeating pose estimation for each person instance. As a result, top-down methods slow down with the increasing number of people (Figure 6). However, compared to bottom-up methods, better accuracies are obtained by top-down approaches.

In this paper, we propose a new bottom-up method, HPRNet, that explicitly handles the hierarchical nature of whole-body pose estimation by regressing keypoints hierarchically. To this end, in addition to the estimation of standard body keypoints, we define the bounding box centers of relatively small body parts such as face and hands with offsets to the person instance center (Figure 3). Concurrently, we build another level of regression where we define each hand and face keypoints with an offset to their corresponding hand and face bounding box centers. We jointly train each level of regression hierarchy and regress all whole-body keypoints with respect to their defined center points. This hierarchical bottom-up approach brings two benefits. First, the scale variance among different body parts are handled naturally as the relative distances within each part are in a similar range and each part-type is processed by a separate sub-network. Second, being a bottom-up method, HPRNet’s inference speed is minimally affected by the number of persons in the input image. This is in contrast to the top-down methods such as ZoomNet, which significantly slows down with more person instances (65.7 ms for an image containing 1 person versus 668.2 ms for an image with 10 persons). Our method is based on the center-point based bottom-up object detection methods [60, 46, 13, 31]. These methods can easily be extended to the keypoint estimation task [60, 47].

We validated the effectiveness of our method through ablation experiments and comparisons with the state of the art (SOTA) on the COCO WholeBody dataset. Our method significantly outperforms all bottom-up methods. It also outperforms the SOTA top-down method ZoomNet in the detection of face and hand keypoints, while being significantly faster than ZoomNet.

Our major contribution in this paper is the proposal of a one-stage, bottom-up method to close the performance gap between the bottom-up and top-down methods. In contrast to top-down methods, our method runs almost in constant

time, independent from the number of persons in the input image.

2. Related Work

2.1. Human Body Pose Estimation

We can categorize the current approaches for multi-person pose estimation into two: bottom-up and top-down. In the bottom-up methods [7, 38, 36, 37, 28, 41, 4, 44, 17, 19, 20, 21, 23, 24], given an image, body keypoints detected first, without knowing the number or location of person instances or to which person instances these keypoints belong. Later, detected keypoints are grouped and assigned to person instances. Recently, center-based object detection methods [60] have been extended to perform human pose estimation [60, 47]. These methods represent keypoints with an offset value to the center of the person box and directly regresses them during training. In order to improve localization of keypoints, they also estimate the heatmap of each keypoint as in other bottom-up methods [6, 36, 41, 28]. At inference, using center offsets, they group and assign keypoints to person instances. Since bottom-up methods detect all people keypoints at once, they are fast.

Top-down methods [8, 42, 16, 15, 50, 54] first detect person instances in the input image. Commonly, they use an off-the-shelf object detector (e.g. FasterRCNN [45]) to obtain person boxes. Next, top-down methods estimate a single person pose for each cropped person box. By cropping and resizing each person box, top-down methods have the advantage to zoom into the details of each person. Therefore, top-down approaches are more capable of handling scale variance issues. As a result, state-of-the-art results are obtained by top-down methods and there is an accuracy gap between top-down and bottom-up approaches. However, since a pose estimation model is run for each person instance, top-down methods tend to be slow on average, that is, they get significantly slower with increasing number of persons in an image (Figure 6).

One may think that using human body pose estimation methods on a whole-body pose estimation dataset (i.e. COCO WholeBody) could be a solution for whole-body pose estimation. However, as it is stated in the COCO WholeBody dataset paper [25], due to the large scale variance between whole body parts, applying these methods directly results in suboptimal accuracies.

2.2. Whole-body Pose Estimation

Whole-body pose estimation requires accurate localization of keypoints on body, face, hand and feet. Detection of keypoints is well studied for each of these body parts independently, under face alignment [5, 53, 52, 59], facial landmark detection [11, 58], hand pose estimation [40, 39], hand tracking [48, 49] and feet keypoint detection [6] top-

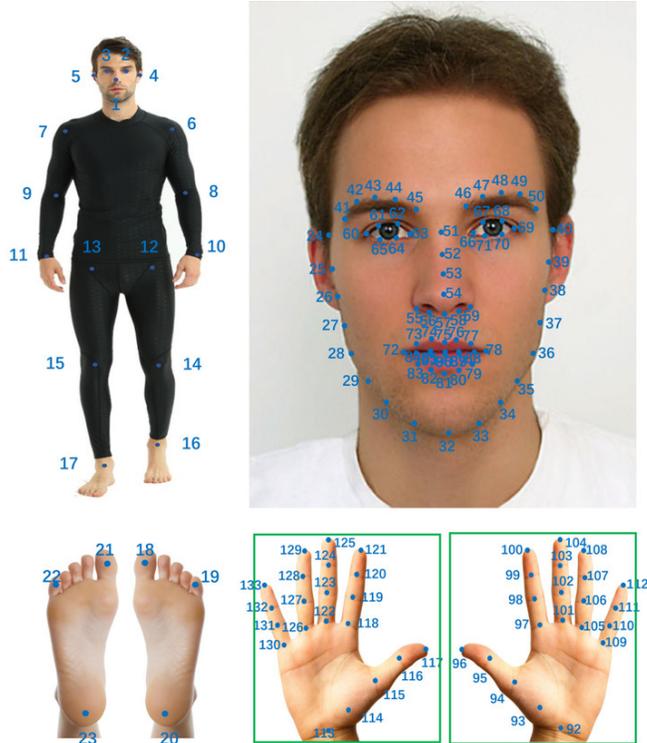


Figure 1: Whole-body keypoints as defined in the COCO WholeBody dataset. There is a total of 133 keypoints. In addition to standard 17 human body keypoints (top-left) from the COCO keypoints dataset, there are 68 face (top-right), 42 hand (21 keypoints for each) (bottom-right) and 6 foot (3 for each) (bottom-left) keypoints are annotated. Image source: <https://github.com/jin-s13/COCO-WholeBody>

ics. However, there are not many works on the whole-body pose estimation mostly due to lack of a large-scale annotated dataset. Prior to the release of the COCO WholeBody dataset [25], OpenPose [6] attempted to detect the whole-body keypoints. For this purpose, OpenPose ensembles 5 separately trained models namely human body pose estimation, hand detection, face detection, hand pose estimation and face pose estimation. Due to these multiple models, training and inference of OpenPose are complex and costly. Our end-to-end trainable single network eliminates these drawbacks.

Hidalgo *et al.* presented a bottom-up method called SN [17]. Their model extends PAF [7] for whole-body pose estimation. Similar to PAF [7], they predict heatmaps for each keypoint and use part affinity maps for grouping. SN model is trained on a dataset that is sampled from different datasets. Both SN and our proposed model HPRNet are bottom-up methods. However, SN falls short of handling scale variations between whole-body parts whereas hierarchical point representation of HPRNet overcomes this issue.

The first step towards having a whole-body pose estimation benchmark is the release of the COCO WholeBody dataset [25]. Jin *et al.* extended the existing COCO key-

points [34] dataset by further annotating face, hands and feet keypoints (Figure 1). They also proposed a strong, two-stage, top-down model to perform whole-body pose estimation on the COCO WholeBody dataset. Similar to top-down human pose estimation methods, Jin *et al.* [25] first obtain candidate person boxes in an image using FasterRCNN [45]. Next, using a single network called ZoomNet, detection of whole-body keypoints is performed on the person boxes. ZoomNet is composed of 4 sub CNN networks. First, FeatureNet processes input person boxes and extracts shared features at two scales. Next, using features from FeatureNet, BodyNet detects body and foot keypoints. BodyNet is also responsible for the prediction of the face and hand bounding box corner points to roughly extract face and hand areas. Later, cropped face and hand bounding boxes are fed to the FaceHead and HandHead networks to detect the keypoints on face and hands. They use HRNet-W32 [50] network for the BodyNet and HRNetV2p-W18 [51] network for the FaceHead and HandHead networks.

Even though bottom-up approaches are fast, they are not robust enough to handle the scale variance across the whole-body parts. However, we hypothesize that representing each

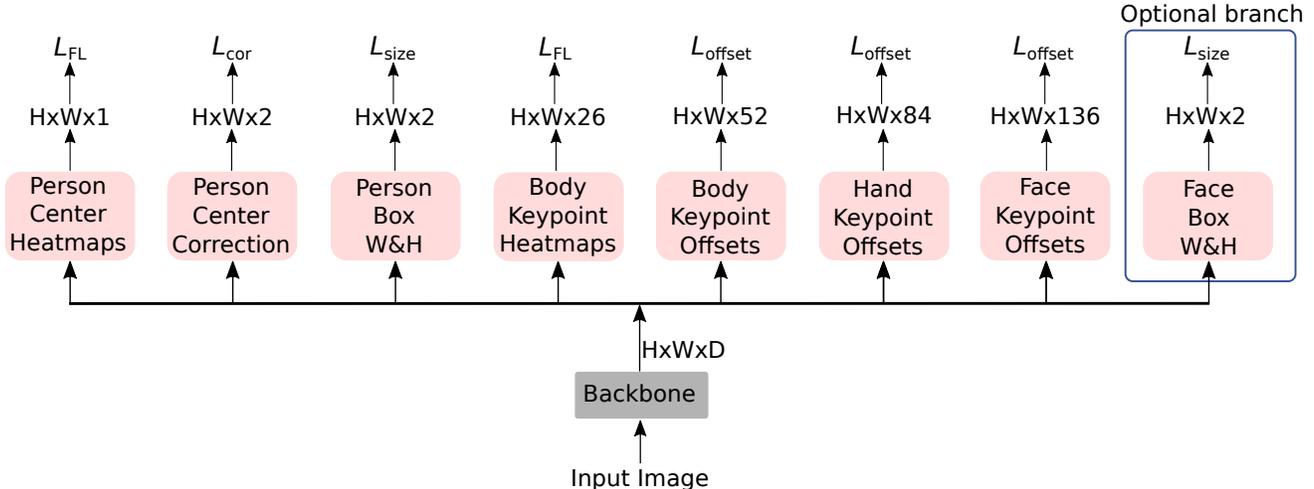


Figure 2: Network architecture of the proposed HPRNet for whole-body keypoint detection.

keypoint with an offset value to a carefully selected location can handle the scale variance. Based on this, we extend the center-based human pose estimation method [60] to perform whole-body pose estimation by introducing hierarchical regression of keypoints. We also show that hierarchical regression of keypoints for small scale whole-body parts (i.e. face and hand) is more effective than cropping and zooming into them.

3. Model

HPRNet is a one-stage end-to-end trainable network that learns regressing the whole-body keypoints. In HPRNet, the input image first passes through a backbone network and output of the backbone is fed to 8 separate branches, namely; *Person Center Heatmap*, *Person Center Correction*, *Person W & H*, *Body Keypoint Offsets*, *Body Keypoint Heatmaps*, *Hand Keypoint Offsets*, *Face Keypoint Offsets* and *Face Box W & H*. We show the network architecture of HPRNet in Figure 2.

3.1. Hierarchical Regression of Whole-Body Keypoints

In HPRNet, we build a hierarchical regression mechanism, where we define each of the whole-body keypoints with a relative location (i.e. offset) to a specific point on the person box.

We represent each of the (standard) 17 keypoints on the body with an offset to the center of the person bounding box. Unlike the body; face, hand and foot are small parts. Based on this, we define each of this parts with a relative location to their part center as follows; (i) each of 68 face keypoints is defined with an offset to the center of face bounding box, (ii) each of 21 left hand keypoints is defined with

an offset to left hand bounding box center, (iii) each of 21 right hand keypoints is defined with an offset to right hand bounding box center, (iv) each of 3 left foot keypoints is defined with an offset to left foot bounding box center, (v) each of 3 right foot keypoints is defined with an offset to right foot bounding box center. Face, hand and foot bounding boxes are automatically extracted from the groundtruth keypoint annotations.

We treat the bounding box center of the face, left hand, right hand, left foot and right foot as a body part keypoint and define each of them with an offset value to the person box center (Figure 3). We illustrate the hierarchical regression of whole-body keypoints in Figure 4b.

At inference, after detecting all the keypoints in the input image, we group and assign them to person instances. To achieve this, we get predicted person centers from the output of *Person Center Heatmap* branch as in CenterNet [60]. Next, we obtain the offset values on the predicted person center locations of *Body Keypoint Offsets* branch output. After that, we add these offsets to person centers to obtain the *regressed* body keypoint locations. At the same time, we extract the *detected* body keypoints from the outputted heatmap of the *Body Keypoint Heatmaps* branch. At the last step, we match the *detected* and *regressed* keypoints based on L2 distance and only take the keypoints inside the predicted person bounding box.

Next, we group face and hand keypoints (and foot keypoints as well, if we are using the Hierarchical Model-I (Figure 4b)). We obtain predicted part centers from the output of *Body Keypoint Heatmaps* branch. Then, we collect the offset values on the corresponding predicted part center locations of *Hand Keypoint Offsets* and *Face Keypoint Offsets* branch output. Finally, we add these offsets to the part centers to obtain the face and hand keypoints.

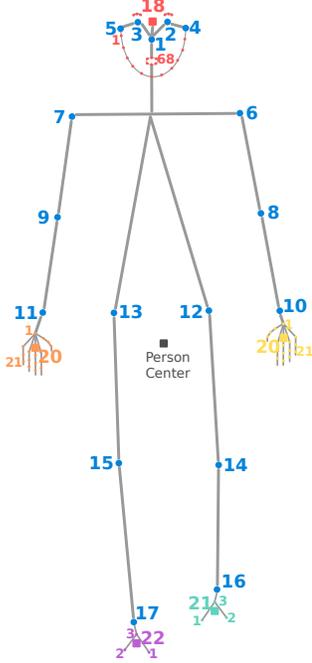


Figure 3: All regressed keypoints in HPRNet. Blue keypoints are body keypoints as defined in COCO keypoints and COCO WholeBody datasets. Colored square points correspond to the face (18), left hand (19), right hand (20), left foot (21) and right foot (22) box centers. Blue keypoints (1-17) and colored square points are defined with an offset to the center of the person instance. For simplicity, face and hand keypoints are sparsely illustrated.

3.2. Regression of Foot Keypoints

Ideally, each labeled foot part in the COCO WholeBody dataset should have 3 keypoint annotations. However, more than 20% of annotated feet have missing annotations (i.e. they have one or two keypoints annotations, instead of three). These missing annotations present a challenge to HPRNet, since we automatically extract foot centers from the annotated extreme points. In the case of the missing foot keypoints, the obtained foot center point is not reliable. To deal with this issue, we treat the foot keypoints as body keypoints as shown in Figure 4c, and represent them by their offsets to the center of the person bounding box.

3.3. Network Architecture

Given an input image of size $4H \times 4W \times 3$, the backbone network outputs a feature map of size $H \times W \times D$. The backbone's output is fed to the following subsequent branches. Each branch has one convolutional layer with 3×3 filters followed by a ReLU layer and another convolutional layer with 1×1 filters.

- *Person Center Heatmap* branch outputs $H \times W$ sized

tensor for person center point predictions.

- *Person Center Correction* branch predicts $H \times W \times 2$ sized tensor for the local offsets of center locations across the spatial axes. These offsets help to recover the lost precision of the center points due to down-sampling operations through the network.
- *Person Box W & H* branch outputs $H \times W \times 2$ sized tensor of widths and heights for each person instance center.
- *Body Keypoint Offsets* branch predicts offset values of 26 keypoints (17 body keypoints + 6 foot keypoints + Center of Face Box + Center of Left Hand Box + Center of Right Hand Box) to the person box center across the x and y axes.
- *Body Keypoint Heatmaps* branch outputs $H \times W \times 26$ sized heatmap tensor for the 26 keypoints.
- *Hand Keypoint Offsets* branch outputs $H \times W \times 84$ sized tensor of offset values between 21 left hand keypoints and left hand box center; and the offset values between the 21 right hand keypoints and right hand box center across the spatial axes.
- *Face Keypoint Offsets* branch outputs $H \times W \times 136$ sized tensor of offset values between 68 face keypoints and face box center across the spatial axes.
- *Face Box W & H* branch outputs $H \times W \times 2$ sized tensor of widths and heights for each face. It is an optional branch.

3.4. Objective Functions

For the optimization of the *Person Center Heatmap (PCH)* and *Body Keypoint Heatmap (BKH)* branches, we use the modified focal loss [33] as done in previous work [31, 61, 60, 46]. Modified focal loss (FL) is presented in Equation 1. $I \in R^{4W \times 4H \times 3}$ is our input image. In HPRNet, due to downsampling operations, the spatial output size of each branch is 4 times smaller resulting in $W \times H$. Therefore, $Y \in [0, 1]^{W \times H \times C}$ is the ground truth heatmap for person centers and keypoints. C corresponds to class number and keypoint types. For instance, in the *Person Center Heatmap* branch, we have only *person* class, thus $C = 1$. $\hat{Y} \in [0, 1]^{W \times H \times C}$ is the predicted heatmap output by the branches where $\hat{Y}_{x,y,c} = 1$ indicates presence of a person center or keypoint at location (x, y) for class c . In the following all equations, N is the total number of ground truth person centers or keypoints in image I . α and β are focal loss parameters and set as $\alpha = 2$ and $\beta = 4$ as in Corner-

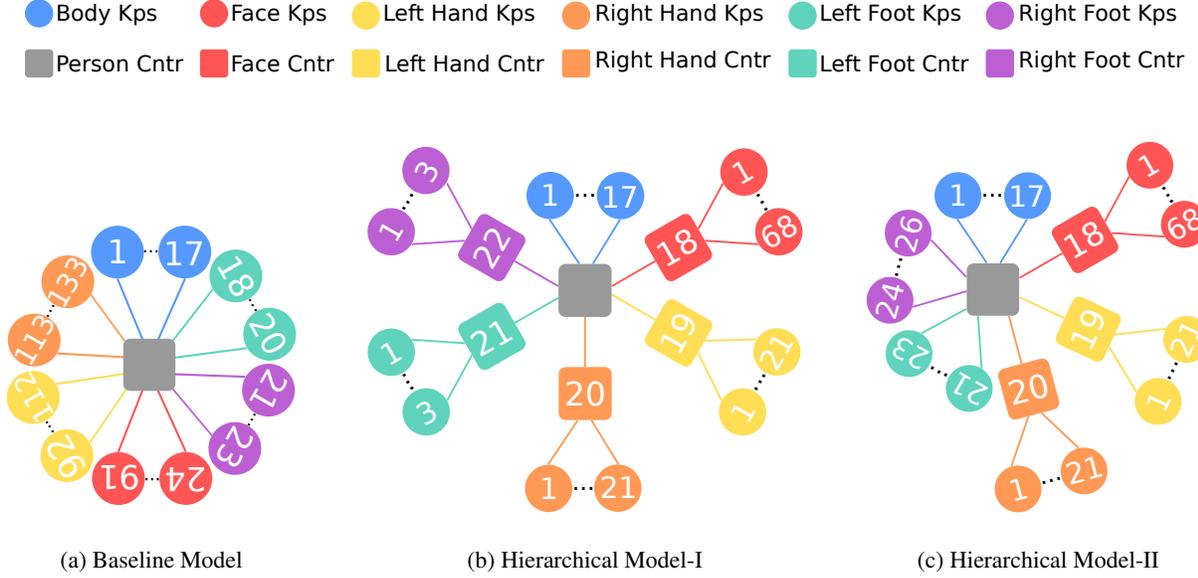


Figure 4: Hierarchical representations of whole-body keypoints. (a) Each of 133 whole-body keypoints is defined with an offset to the person box center. (b) Body keypoints and other part centers (i.e. foot, face and hand) are defined with offsets to the person box center. Foot, face and hand keypoints are defined with offsets according to their corresponding part centers. (c) Considering the sparsity of foot keypoint annotations, we define them with their offset values to the person box center. In both Hierarchical Model-I and Hierarchical Model-II, body keypoints are defined with offset values to the person box center. Each face and hand keypoint is defined with an offset to face and hand bounding box centers, respectively.

Table 1: Comparison of Hierarchical Model-I (HM-I) and Hierarchical Model-II (HM-II) as in Figure 4. Training foot keypoints with offset values to the person box center outperforms the model when trained with offset values to the foot part centers. Both models are trained with DLA backbone.

Method	body		foot		face		hand		whole-body		all-mean	
	AP^{kp}	AR^{kp}										
HM-I	55.5	63.4	33.5	55.3	74.6	83.5	44.1	57.8	28.0	40.5	47.1	60.1
HM-II	55.2	63.1	49.1	60.9	74.6	83.7	47.0	60.8	31.5	44.6	51.5	62.6

Net [31].

$$L_{FL} = \frac{-1}{N} \sum_{xy_c} \begin{cases} (1 - \hat{Y}_{xy_c})^\alpha \log(\hat{Y}_{xy_c}) & \text{if } Y_{xy_c} = 1 \\ (1 - Y_{xy_c})^\beta (\hat{Y}_{xy_c})^\alpha & \\ \log(1 - \hat{Y}_{xy_c}) & \text{otherwise} \end{cases} \quad (1)$$

To compensate for the discretization error of the person center points due to down-sampling operations through the network, we optimize the *Person Center Correction* according to the following L1 loss similar to the bottom-up object detectors [31, 61, 60, 46]. $\hat{T} \in \mathcal{R}^{W \times H \times 2}$ is the predicted local offset by the network to recover the lost precision of person center points. $p \in \mathcal{R}^2$ is a ground truth keypoint and $\tilde{p} = \lfloor \frac{p}{4} \rfloor$ is the corresponding ground keypoint location at

low-resolution.

$$L_{cor} = \frac{1}{N} \sum_p \left| \hat{T}_{\tilde{p}} - \left(\frac{p}{4} - \tilde{p} \right) \right| \quad (2)$$

We optimize the *Body Keypoint Offset*, *Hand Keypoint Offset* and *Face Keypoint Offset* branches using the L1 loss. The generic formulation of keypoint regression is presented in Equation 3. In the equation, $\hat{O} \in \mathbb{R}^{H \times W \times k \times 2}$ is the regression output of keypoints k for a specific whole-body part (i.e. body, face, hand), and B_{part} is the ground truth center of that part’s bounding box.

$$L_{offset} = \sum_k \left| \hat{O}_{[k]} - B_{part} \right| \quad (3)$$

Finally, for the *Person Box H & W* and *Face Box H & W* branches, we use L1 loss and scale it by 0.1 as in

Table 2: Comparing HPRNet with the baseline model. To obtain the baseline results, we regress all the 133 keypoints to the person instance box center during training as in Figure 4b. Both models are trained with DLA backbone.

Method	body		foot		face		hand		whole-body		all-mean	
	AP^{kp}	AR^{kp}										
Baseline	46.7	55.5	33.6	48.9	52.0	60.2	26.4	39.1	33.3	43.4	38.4	49.4
HPRNet	55.2	63.1	49.1	60.9	74.6	83.7	47.0	60.8	31.5	44.6	51.5	62.6

Table 3: Comparison with the state-of-the-art on COCO WholeBody validation set. The methods are divided into two groups: top-down and bottom-up. The best results and run times are boldfaced separately for each group. HPRNet performs best among the bottom-up methods. HPRNet also obtains state-of-the-art results on face and hand keypoint detection outperforming ZoomNet. Among all methods, HPRNet with DLA backbone is the fastest one. * indicates that run time linearly increases as the number of people in an image increases. HG is Hourglass-104. R. time is Running time.

Method	body		foot		face		hand		whole-body		all-mean		R. time (ms)
	AP^{kp}	AR^{kp}											
<i>Top-down methods:</i>													
OpenPose [6]	56.3	61.2	53.2	64.5	48.2	62.6	19.8	34.2	33.8	44.9	42.3	53.5	45
HRNet* [50]	65.9	70.9	31.4	42.4	52.3	58.2	30.0	36.3	43.2	52.0	44.6	52.0	-
ZoomNet* [25]	74.3	80.2	79.8	86.9	62.3	70.1	40.1	49.8	54.1	65.8	62.1	70.6	175
<i>Bottom-up methods:</i>													
PAF [7]	26.6	32.8	10.0	25.7	30.9	36.2	13.3	32.1	14.1	18.5	19.0	29.1	100
SN [17]	28.0	33.6	12.1	27.7	38.2	44.0	13.8	33.6	16.1	20.9	21.6	32.0	216
AE [36]	40.5	46.4	7.7	16.0	47.7	58.0	34.1	43.5	27.4	35.0	31.5	39.8	-
Ours (HPRNet-DLA)	55.2	63.1	49.1	60.9	74.6	83.7	47.0	60.8	31.5	44.6	51.5	62.6	37
Ours (HPRNet-HG)	59.4	68.3	53.0	65.4	75.4	86.8	50.4	64.2	34.8	49.2	54.6	66.8	101

CenterNet [60]. In the Equation 4, $s_n = (w, h)$ is the width and height values of the each object (or face) n and $\hat{S} \in \mathcal{R}^{W \times H \times 2}$ is the predicted width and height values.

$$L_{\text{size}} = \frac{1}{N} \sum_{n=1}^N \left| \hat{S}_{p_n} - s_n \right| \quad (4)$$

We obtain the overall loss by summing the losses from all branches as follows:

$$L_{\text{overall}} = L_{\text{FL}}^{\text{PCH}} + L_{\text{FL}}^{\text{BKH}} + L_{\text{cor}} + L_{\text{offset}}^{\text{body}} + L_{\text{offset}}^{\text{face}} + L_{\text{offset}}^{\text{hand}} + 0.1L_{\text{size}}^{\text{person}} + 0.1L_{\text{size}}^{\text{face}} \quad (5)$$

4. Experiments

This section describes the experiments we conducted to show the effectiveness of our proposed method. First, we present ablation experiments to compare hierarchical models I and II shown in Figure 4. Next, we compare our method with our baseline CenterNet [60] (Figure 4a). Finally, we provide performance comparison with the state of the art and a run-time analysis.

Implementation Details We use Deep Layer Aggregation (DLA) [57] backbone for ablation and baseline com-

parison experiments, and Hourglass-104 [31] as our backbone network for state of the art comparison. For all experiments, during training we resize the images to 512×512 pixels. At inference we use images with their original sizes without applying any scaling. We train all the models with a batch size of 32 for 140 epochs using the Adam optimizer [27]. We set the initial learning rate to 1.25×10^{-4} and divided it by 10 at epochs 90 and 120. We trained all of the models on 4 Tesla V100 GPUs, and tested using a single GTX 1080 TI GPU. We used PyTorch [43] to implement our models. All of our experiments are conducted on the COCO WholeBody Dataset [25] and results are presented in keypoint AP (AP^{kp}) and keypoint recall AR (AR^{kp}) metrics without any test time augmentation. All results are obtained on the COCO WholeBody validation set.

4.1. Hierarchical Model-I vs Hierarchical Model-II

In Table 1, we compare Hierarchical Model-I and Hierarchical Model-II (see Figure 4). As it can be seen from the table, regressing foot keypoints as a part of the body keypoints, improves the foot AP^{kp} significantly by 15.6 points (33.5 vs. 49.1). Moreover, hand and whole-body AP^{kp} s also improved about 3 points in this setup. Based on these results, for the rest of the experiments we use the Hierarchical Model-II.

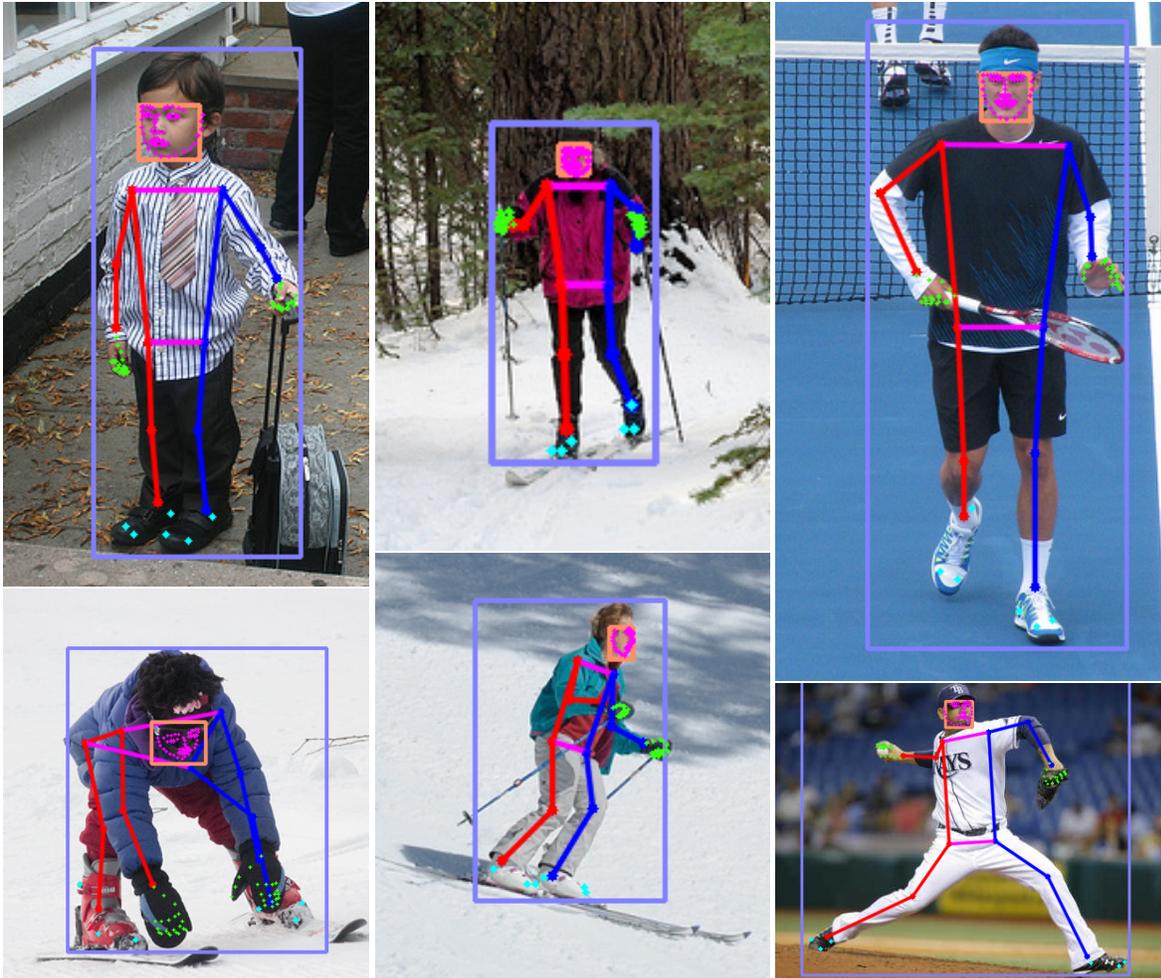


Figure 5: Sample whole-body keypoint detection results of HPRNet. We show correctly detected people, and their whole-body poses. Detection box is marked with a purple bounding box, and body pose estimation is shown with blue color for the left parts, red color for the right parts. For clarity, we mark the detected keypoints on face, hand and foot with magenta, green and cyan colors, respectively. Detected faces are marked with an orange bounding box.

4.2. Comparison with Baseline

To obtain the baseline results, we regress all the 133 keypoints to the person instance box center during training as in CenterNet [60] (see Figure 4b). In Table 2, we compare HPRNet with the baseline model in terms of accuracy and recall. Our proposed HPRNet significantly outperforms the baseline results for all AP^{kp} and AR^{kp} metrics except the whole-body AP^{kp} .

4.3. Comparison with the State-of-the-art

Table 3 presents the performance of our models and several established keypoint estimation models on the COCO WholeBody validation set. We also present average run times if available. **HPRNet performs best among the bottom-up methods.** Other bottom-up methods especially

fail to accurately localize foot keypoints. The performance gap between the second best performing bottom-up method and our method is 40.9 AP^{kp} points on the foot keypoint detection. Similarly, our method outperforms other bottom-up methods for the body, face, hand and whole-body keypoint detection by a large margin. Among the top-down methods, ZoomNet outperforms the well known OpenPose [6] and HRNet [50]. Here, ZoomNet is a two-stage framework where at the first stage person candidates are extracted with FasterRCNN and at the second stage ZoomNet is run on these candidate boxes. HRNet can be seen as a one-stage counterpart of ZoomNet and finally OpenPose is a multi-model which requires separate training for each whole-body part. HPRNet obtains state-of-the-art results on face and hand keypoint detection. Our model with Hourglass-104 backbone outperforms ZoomNet on the detection of face

Table 4: Face detection results. The first group of results are obtained from extreme face keypoints for both ZoomNet and HPRNet. The HPRNet results in the second group are obtained with an extra face detection branch. HG is Hourglass-104.

Method	AP	AP ₅₀	AP ₇₅	AP _M	AP _L
<i>When face boxes are extracted from extreme face keypoints:</i>					
ZoomNet	37.7	64.5	41.1	25.8	44.9
HPRNet (DLA)	46.2	70.6	54.8	32.2	53.8
HPRNet (HG)	46.1	70.9	53.6	33.4	53.1
<i>Our model with an extra face detection branch</i>					
HPRNet (DLA)	55.8	82.3	66.2	40.0	63.6
HPRNet (HG)	56.4	82.4	67.1	43.4	63.3

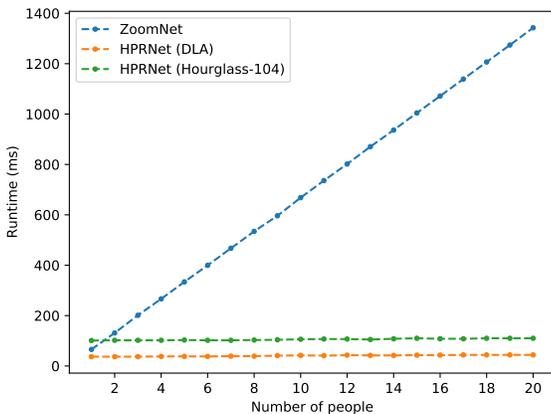


Figure 6: Runtime analysis of ZoomNet and our models with respect to number of people in an image. As the number of people in an image increases, the runtime of ZoomNet linearly increases. Whereas, our models almost have constant run time.

keypoints by 13.1 AP^{kp} points and hand keypoints by 10.3 AP^{kp} points. These successful results on the face and hand keypoint detection, further shows the effectiveness of our proposed bottom-up hierarchical approach over the ZoomNet’s zoom-in mechanism. However, for the detection of the body and whole-body keypoints ZoomNet performs best among all methods. **Among all methods, our HPRNet with the DLA backbone is the fastest one (37 ms) with constant run time.** In Figure 5, we show sample qualitative results for our approach.

Runtime Analysis Average run time of ZoomNet (including Faster RCNN for person detector) on a single image is 174.7 ms. Similarly, the average run time of HPRNet with DLA and Hourglass-104 backbones is 37 ms (26 ms for feedforward and 11 ms for keypoint grouping and assignment) and 101 ms (90 ms for feedforward and 11 ms

for keypoint grouping and assignment). HPRNet is significantly faster than ZoomNet. Moreover, as a top-down method, run time of ZoomNet increases as the number of people on an image increases. We compare the run time of our models and ZoomNet in Figure 6.

4.4. Face Detection from Keypoints

In this section, we studied the face detection task and compared HPRNet and ZoomNet. We first extracted face boxes using extreme face keypoints and calculated AP scores as in object detection. Our model outperformed ZoomNet in face detection (46.2 AP vs 37.7 AP). Later, using an additional branch for face detection we train another model (see Figure 2). Our model with an extra face detection branch further improved the performance of HPRNet for face detection achieving 55.8 AP and 56.4 AP with DLA and Hourglass-104 backbones respectively. Results are presented in Table 4.

5. Conclusion

In this work, we introduced HPRNet as a bottom-up, one-stage method for whole-body keypoint detection. HPRNet handles scale variance among whole-body parts by hierarchically regressing whole-body keypoints. We evaluated the effectiveness of our method through baseline comparison and ablation experiments on hierarchical structure of whole-body keypoints. Our method achieves state-of-the-art results in the detection of face and hand keypoints on the COCO WholeBody dataset; it also outperforms all other bottom-up methods in the detection of all whole-body parts. We conducted a run time analysis between HPRNet and ZoomNet and showed that in contrast to ZoomNet, HPRNet runs in constant time, independent of the number of persons in an image.

6. Acknowledgements

The numerical calculations reported in this paper were fully performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

References

- [1] 3d skeletal tracking on azure kinect.
- [2] Azure kinect body tracking joints.
- [3] How huawei ml kit's face detection and hand keypoint detection capabilities helped with creating the game crazy rockets.
<https://medium.com/huawei-developers/how-huawei-ml-kits-face-detection-and-hand-keypoint-detection-capabilities-helped-with-creating-6a22fdb7f967>.
- [4] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, 2016.
- [5] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 2014.
- [6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [9] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, 2020.
- [10] Gokcen Cimen, Christoph Maurhofer, Bob Sumner, and Martin Guay. Ar poser: Automatically augmenting mobile pictures with digital avatars imitating poses. In *12th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing*, 2018.
- [11] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [12] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [13] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *IEEE International Conference on Computer Vision*, 2019.
- [14] Ahmed Elhayek, Onorina Kovalenko, Pramod Murthy, Jameel Malik, and Didier Stricker. Fully automatic multi-person human motion capture for vr applications. In *International Conference on Virtual Reality and Augmented Reality*, 2018.
- [15] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *IEEE International Conference on Computer Vision*, 2017.
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE International Conference on Computer Vision*, 2017.
- [17] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [18] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Part-aligned pose-guided recurrent network for action recognition. *Pattern Recognition*, 2019.
- [19] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [20] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepcut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, 2016.
- [21] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [22] Umar Iqbal, Kevin Xie, Yunrong Guo, Jan Kautz, and Pavlo Molchanov. Kama: 3d keypoint aware body mesh articulation. *arXiv preprint arXiv:2104.13502*, 2021.
- [23] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [24] Sheng Jin, Xujie Ma, Zhipeng Han, Yue Wu, Wei Yang, Wentao Liu, Chen Qian, and Wanli Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, 2017.
- [25] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *European Conference on Computer Vision*, 2020.
- [26] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [28] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *European Conference on Computer Vision*, 2018.
- [29] Laxman Kumarapu and Prerana Mukherjee. Animepose: Multi-person 3d pose estimation and animation. *Pattern Recognition Letters*, 2021.
- [30] Jogendra Nath Kundu, Mugalodi Rakesh, Varun Jampani, Rahul Mysore Venkatesh, and R Venkatesh Babu. Appearance consensus driven self-supervised human mesh recovery. In *European Conference on Computer Vision*, 2020.

- [31] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision*, 2018.
- [32] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [33] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017.
- [34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.
- [35] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [36] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *arXiv preprint arXiv:1611.05424*, 2016.
- [37] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016.
- [38] Guanghan Ning, Zhi Zhang, and Zhiqian He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE Transactions on Multimedia*, 2017.
- [39] Markus Oberweger and Vincent Lepetit. Deeprior++: Improving fast and accurate 3d hand pose estimation. In *IEEE International Conference on Computer Vision Workshops*, 2017.
- [40] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *20th Computer Vision Winter Workshop*, 2015.
- [41] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *European Conference on Computer Vision*, 2018.
- [42] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 2019.
- [44] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.
- [46] Nermin Samet, Samet Hicsonmez, and Emre Akbas. Houghnet: Integrating near and long-range evidence for bottom-up object detection. In *European Conference on Computer Vision*, 2020.
- [47] Nermin Samet, Samet Hicsonmez, and Emre Akbas. Houghnet: Integrating near and long-range evidence for visual detection. 2021.
- [48] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Lichten, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015.
- [49] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [50] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [51] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.
- [52] George Trigeorgis, Patrick Snape, Mihalis A Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [53] Georgios Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [54] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, 2018.
- [55] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo2cap2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE transactions on visualization and computer graphics*, 2019.
- [56] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [57] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [58] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.
- [59] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning deep representation for face alignment with

auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

- [60] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.
- [61] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.