Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression

Zigang Geng[1,3*], Ke Sun[1*], Bin Xiao[3], Zhaoxiang Zhang[2], Jingdong Wang[3†]
[1]University of Science and Technology of China
[2]Institute of Automation, CAS, University of Chinese Academy of Sciences
Centre for Artificial Intelligence and Robotics, HKISI_CAS
[3]Microsoft
{zigang,sunk}@mail.ustc.edu.cn, zhaoxiang.zhang@ia.ac.cn, {bixi,jingdw}@microsoft.com

Abstract

In this paper, we are interested in the bottom-up paradigm of estimating human poses from an image. We study the dense keypoint regression framework that is previously inferior to the keypoint detection and grouping framework. Our motivation is that regressing keypoint positions accurately needs to learn representations that focus on the keypoint regions.

We present a simple yet effective approach, named disentangled keypoint regression (DEKR). We adopt adaptive convolutions through pixel-wise spatial transformer to activate the pixels in the keypoint regions and accordingly learn representations from them. We use a multi-branch structure for separate regression: each branch learns a representation with dedicated adaptive convolutions and regresses one keypoint. The resulting disentangled representations are able to attend to the keypoint regions, respectively, and thus the keypoint regression is spatially more accurate. We empirically show that the proposed direct regression method outperforms keypoint detection and grouping methods and achieves superior bottom-up pose estimation results on two benchmark datasets, COCO and CrowdPose. The code and models are available at https://github.com/HRNet/DEKR.

1. Introduction

Human pose estimation is a problem of predicting the keypoint positions of each person from an image, i.e., localize the keypoints as well as identify the keypoints belonging to the same person. There are broad applications, including action recognition, human-computer interaction, smart photo editing, pedestrian tracking, etc.

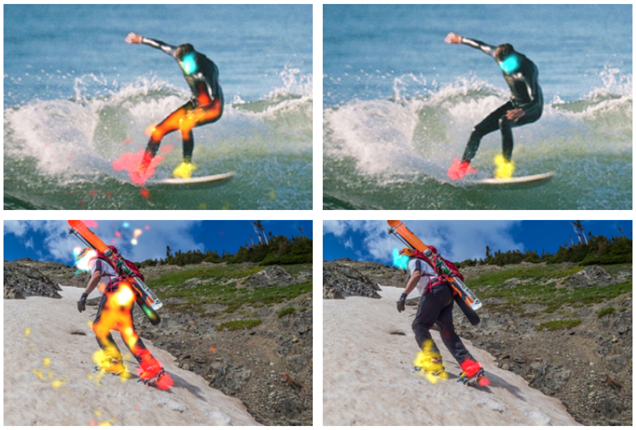
Figure 1. Illustration of the salient regions for regressing the keypoints. We take three keypoints, nose and two ankles, as an example for illustration clarity. Left: baseline. Right: our approach DEKR. It can be seen that our approach is able to focus on the keypoint regions. The salient regions are generated using the tool [46].

There are two main paradigms: top-down and bottom-up. The top-down paradigm first detects the person and then performs single-person pose estimation for each detected person. The bottom-up paradigm either directly regresses the keypoint positions belonging to the same person, or detects and groups the keypoints, such as affinity linking [7, 31], associative embedding [40], HGG [27] and HigherHRNet [11]. The top-down paradigm is more accurate but more costly due to an extra person detection process, and the bottom-up paradigm, the interest of this paper, is more efficient.

The recently-developed pixel-wise keypoint regression approach, CenterNet [78], estimates the K keypoint positions together for each pixel from the representation at the pixel. Direct regression to keypoint positions in CenterNet [78] performs reasonably. But the regressed keypoints are spatially not accurate and the performance is worse than the keypoint detection and grouping scheme. Figure 1 (left) shows two examples in which the salient areas for keypoint

[*]This work was done when Zigang Geng and Ke Sun were interns at Microsoft Research, Beijing, P.R. China
[†]Corresponding author

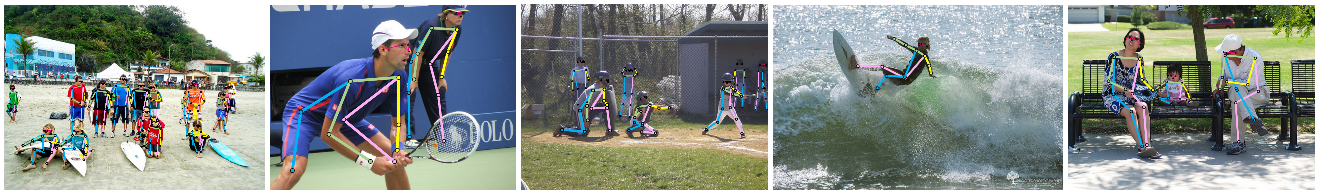

Figure 2. Multi-person pose estimation. The challenges include diverse person scales and orientations, various poses, etc. Example results are from our approach DEKR.

regression spread broadly and the regression quality is not satisfactory.

We argue that regressing the keypoint positions accurately needs to learn representations that focus on the keypoint regions. Starting from this *regression by focusing* concept, we present a simple yet effective approach, named disentangled keypoint regression (DEKR). We adopt adaptive convolutions, through pixel-wise spatial transformer (a pixel-wise extension of spatial transformer network [26]), to activate the pixels lying in the keypoint regions, and then learn the representations from these activated pixels, so that the learned representations can focus on the keypoint regions.

We further decouple the representation learning for one keypoint from other keypoints. We adopt a separate regression scheme through a multi-branch structure: each branch learns a representation for one keypoint with adaptive convolutions dedicated for the keypoint and regresses the position for the corresponding keypoint. Figure 1 (right) illustrates that our approach is able to learn highly concentrative representations, each of which focuses on the corresponding keypoint region.

Experimental results demonstrate that the proposed DEKR approach improves the localization quality of the regressed keypoint positions. Our approach, that performs direct keypoint regression without matching the regression results to the closest keypoints detected from the keypoint heatmaps, outperforms keypoint detection and grouping methods and achieves superior performance over previous state-of-the-art bottom-up pose estimation methods on two benchmark datasets, COCO and CrowdPose.

Our contributions to bottom-up human pose estimation are summarized as follows.

- We argue that the representations for regressing the positions of the keypoints accurately need to focus on the keypoint regions.

- The proposed DEKR approach is able to learn disentangled representations through two simple schemes, adaptive convolutions and multi-branch structure, so that each representation focuses on one keypoint region and the prediction of the corresponding keypoint position from such representation is accurate.

- The proposed direct regression approach outperforms keypoint detection and grouping schemes and achieves new state-of-the-art bottom-up pose estimation results on the benchmark datasets, COCO and CrowdPose.

2. Related Work

The convolutional neural network (CNN) solutions [17, 35, 56, 70, 43, 45, 49, 74, 54] to human pose estimation have shown superior performance over the conventional methods, such as the probabilistic graphical model or the pictorial structure model [72, 50]. Early CNN approaches [62, 2, 8] directly predict the keypoint positions for single-person pose estimation, which is later surpassed by the heatmap estimation based methods [5, 20, 13, 37, 1]. The geometric constraints and structured relations among body keypoints are studied for performance improvement [12, 71, 9, 60, 28, 75].

Top-down paradigm. The top-down methods perform single-person pose estimation by firstly detecting each person from the image. Representative works include: HRNet [57, 66], PoseNet [48], RMPE [18], convolutional pose machine [68], Hourglass [41], Mask R-CNN [21], CFN [23], Integral pose regression [58], CPN [10], simple baseline [69], CSM-SCARB [55], Graph-PCNN [65], RSN [6], and so on. These methods exploit the advances in person detection as well as extra person bounding-box labeling information. The top-down paradigm, though achieving satisfactory performance, takes extra cost in person box detection.

Other developments include improving the keypoint localization from the heatmap [22, 73], refining pose estimation [19, 39], better data augmentation [4], developing a multi-task learning architecture combining detection, segmentation and pose estimation [30], and handling the occlusion issue [34, 52, 77].

Bottom-up paradigm. Most existing bottom-up methods mainly focus on how to associate the detected keypoints that belong to the same person together. The pioneering work, DeepCut [51], DeeperCut [24], and L-JPA [25] formulate the keypoint association problem as an integer linear program, which however takes longer processing time (e.g., the order of hours).

Various grouping techniques are developed, such as part-affinity fields in OpenPose [7] and its extension in Pif-

Paf [31], associative embedding [40], greedy decoding with hough voting in PersonLab [47], and graph clustering in HGG [27].

Several recent works [78, 44, 42, 67] densely regress a set of pose candidates, where each candidate consists of the keypoint positions that might be from the same person. Unfortunately, the regression quality is not high, and the localization quality is weak. A post-processing scheme, matching the regressed keypoint positions to the closest keypoints (which is spatially more accurate) detected from the keypoint heatmaps, is usually adopted to improve the regression results.

Our approach aims to improve the direct regression results, by exploring our *regression by focusing* idea. We learn K disentangled representations, each of which is dedicated for one keypoint and learns from the adaptively activated pixels, so that each representation focuses on the corresponding keypoint area. As a result, the position prediction for one keypoint from the corresponding disentangled representation is spatially accurate. Our approach is superior to and differs from [63] that uses the mixture density network for handling uncertainty to improve direct regression results.

Disentangled representation learning. Disentangled representations [3] have widely been studied in computer vision [38, 15, 76, 64, 79], e.g., disentangling the representations into content and pose [15], disentangling motion from content [64], disentangling pose and appearance [76].

Our proposed disentangled regression in some sense can be regarded as disentangled representation learning: learn the representation for each keypoint separately from the corresponding keypoint region. The idea of representation disentanglement for pose estimation is also explored in the top-down approach, part-based branching network (PBN) [59], which learns high-quality heatmaps by disentangling representations into each part group. They are clearly different: our approach learns representations focusing on each keypoint region for position regression, and PBN de-correlates the appearance representations among different part groups.

3. Approach

Given an image I, multi-person pose estimation aims to predict the human poses, where each pose consists of K keypoints, such as shoulder, elbow, and so on. Figure 2 illustrates the multi-person pose estimation problem.

3.1. Disentangled Keypoint Regression

The pixel-wise keypoint regression framework estimates a candidate pose at each pixel $\mathbf{q}$ (called center pixel), by predicting an $2K$-dimensional offset vector $\mathbf{o}_q$ from the center pixel $\mathbf{q}$ for the K keypoints. The offset maps O, containing the offset vectors at all the pixels, are estimated through a keypoint regression head,

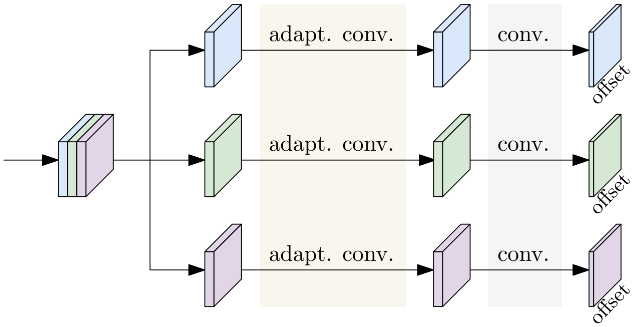

Figure 3. Disentangled keypoint regression. Each branch learns the representation for one keypoint through two adaptive convolutions from a partition of the feature maps output from the backbone and regresses the 2D offset of each keypoint using a 1×1 convolution separately. This is an illustration for three keypoints, and the feature maps are divided into three partitions, each fed into one branch. In our experiments on COCO pose estimation, the feature maps are divided into 17 partitions and there are 17 branches for regressing the 17 keypoints.

$$\mathsf{O} = \mathcal{F}(\mathsf{X}), \tag{1}$$

where X is the feature computed from a backbone, HRNet in this paper, and $\mathcal{F}(\ )$ is the keypoint position regression head predicting the offset maps O.

The structure of the proposed disentangled keypoint regression (DEKR) head is illustrated in Figure 3. DEKR adopts the multi-branch parallel adaptive convolutions to learn disentangled representations for the regression of the K keypoints, so that each representation focuses on the corresponding keypoint region.

Adaptive activation. One normal convolution (e.g., 3×3 convolution) only sees the pixels nearby the center pixel $\mathbf{q}$. A sequence of several normal convolutions may see the pixels farther from the center pixel that might lie in the keypoint region, but might not focus on and highly activate these pixels.

We adopt the adaptive convolutions, to learn representations focusing on the keypoint region. The adaptive convolution is a modification of a normal convolution (e.g., 3×3 convolution):

$$\mathbf{y}(\mathbf{q}) = \sum\nolimits_{i=1}^{9} \mathbf{W}_i \mathbf{x}(\mathbf{g}_{si}^q + \mathbf{q}).$$

Here, $\mathbf{q}$ is the center (2D) position, and $\mathbf{g}_{si}^q$ is the offset, $\mathbf{g}_{si}^q + \mathbf{q}$ corresponds to the ith activated pixel. $\{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_9\}$ are the kernel weights.

The offsets $\{\mathbf{g}_{s1}^q, \mathbf{g}_{s2}^q, \dots, \mathbf{g}_{s9}^q\}$ (denoted by a 2×9 matrix $\mathbf{G}_s^q$) can be estimated by an extra normal 3×3 convolution in a nonparametric way like deformable convolutions [14], or in a parametric way extending the spatial transformer network [26] from a global manner to a pixel-wise manner. We adopt the latter one and estimate an affine transformation matrix $\mathbf{A}^q$ ($\in \mathbb{R}^{2\times 2}$) and a translation vector $\mathbf{t}$ ($\in \mathbb{R}^{2\times 1}$) for each pixel. Then $\mathbf{G}_s^q = \mathbf{A}^q \mathbf{G}_t + [\mathbf{t}\ \mathbf{t}\ \dots\ \mathbf{t}]$. $\mathbf{G}_t$ represents the regular 3×3 position (meaning that a normal convolution is conducted in the transformed space),

$$\mathbf{G}_t = \begin{bmatrix} -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Separate regression. The offset regressor $\mathcal{F}$ in [78] is a single branch, and estimates all the K $2D$ offsets together from a single feature for each position. We propose to use a K-branch structure, where each branch performs the adaptive convolutions and then regresses the offset for the corresponding keypoint.

We divide the feature maps X output from the backbone into K feature maps, $\mathsf{X}_1, \mathsf{X}_2, \dots, \mathsf{X}_K$, and estimate the offset map O_k for each keypoint from the corresponding feature map:

$$\mathsf{O}_1 = \mathcal{F}_1(\mathsf{X}_1) \tag{2}$$
$$\mathsf{O}_2 = \mathcal{F}_2(\mathsf{X}_2) \tag{3}$$
$$\vdots$$
$$\mathsf{O}_K = \mathcal{F}_K(\mathsf{X}_K), \tag{4}$$

where $\mathcal{F}_k(\ )$ is the kth regressor on the kth branch, and O_k is the offset map for the kth keypoint. The K regressors, $\{\mathcal{F}_1(\ ), \mathcal{F}_2(\ ), \dots, \mathcal{F}_K(\ )\}$ have the same structures, and their parameters are learned independently.

Each branch in separate regression is able to learn its own adaptive convolutions, and accordingly focuses on activating the pixels in the corresponding keypoint region (see Figure 4 (b - e)). In the single-branch case, the pixels around all the keypoints are activated, and the activation is not focused (see Figure 4 (a)).

The multi-branch structure explicitly decouples the representation learning for one keypoint from other keypoints, and thus improves the regression quality. In contrast, the single-branch structure has to decouple the feature learning implicitly which increases the optimization difficulty. Our results in Figure 5 show that the multi-branch structure reduces the regression loss.

3.2. Loss Function

Regression loss. We use the normalized smooth loss to form the pixel-wise keypoint regression loss:

$$\ell_p = \sum\nolimits_{i \in \mathcal{C}} \frac{1}{Z_i} \operatorname{smooth}_{L_1}(\mathbf{o}_i - \mathbf{o}_i^*). \tag{5}$$

Here, $Z_i = \sqrt{H_i^2 + W_i^2}$ is the size of the corresponding person instance, and H_i and W_i are the height and the width of the instance box. $\mathcal{C}$ is the set of the positions that have groundtruth poses. $\mathbf{o}_i$ ($\mathbf{o}_i^*$), a column of the offset maps O (O^*), is the $2K$-dimensional estimated (groundtruth) offset vector for the position i.

Keypoint and center heatmap estimation loss. We also estimate K keypoint heatmaps each corresponding to a keypoint type and the center heatmap indicating the confidence that each pixel is the center of some person, using a separate heatmap estimation branch,

$$(\mathsf{H}, \mathbf{C}) = \mathcal{H}(\mathsf{X}). \tag{6}$$

The heatmaps are used for scoring and ranking the regressed poses. The heatmap estimation loss function is formulated as the weighted distances between the predicted heat values and the groundtruth heat values:

$$\ell_h = \|\mathsf{M}^h \odot (\mathsf{H} - \mathsf{H}^*)\|_2^2 + \|\mathbf{M}^c \odot (\mathbf{C} - \mathbf{C}^*)\|_2^2. \tag{7}$$

Here, $\|\cdot\|_2$ is the entry-wise 2-norm. $\odot$ is the element-wise product operation. M^h has K masks, and the size is $H \times W \times K$. The kth mask, M_k^h, is formed so that the mask weight of the positions not lying in the kth keypoint region is 0.1, and others are 1. The same is done for the mask $\mathbf{M}^c$ for the center heatmap. H^* and $\mathbf{C}^*$ are the target keypoint and center heatmaps.

Whole loss. The whole loss function is the sum of the heatmap loss and the regression loss:

$$\ell = \ell_h + \lambda \ell_p, \tag{8}$$

where λ is a trade-off weight, and set as 0.03 in our experiments.

3.3. Inference

A testing image is fed into the network, outputting the regressed pose at each position, and the keypoint and center heatmaps. We first perform the center NMS process on the center heatmap to remove non-locally maximum positions and the positions whose center heat value is not higher than 0.01. Then we perform the pose NMS process over the regressed poses at the positions remaining after center NMS, to remove some overlapped regressed poses, and maintain at most 30 candidates. The score used in pose NMS is the average of the heat values at the regressed K keypoints, which is helpful to keep candidate poses with highly accurately localized keypoints.

We rank the remaining candidate poses using the score that is estimated by jointly considering their corresponding center heat values, keypoint heat values and their shape scores. The shape feature includes the distance and the relative offset between a pair of neighboring keypoints[1]: $\{d_{ij} | (i,j) \in \mathcal{E}\}$ and $\{\mathbf{p}_i - \mathbf{p}_j | (i,j) \in \mathcal{E}\}$, and keypoint heat values indicating the visibility of each keypoint. We

[1] A neighboring pair (i, j) corresponds to a stick in the COCO dataset, and there are 19 sticks (denoted by $\mathcal{E}$) in the COCO dataset.

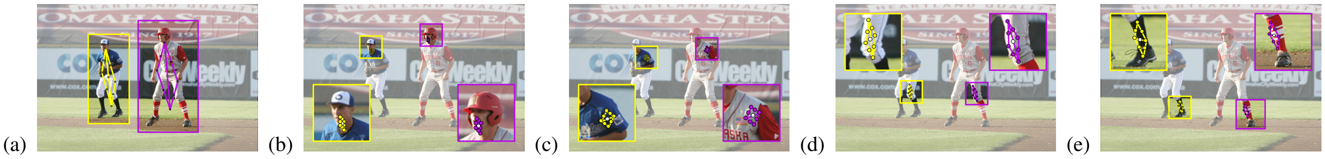

Figure 4. Illustrating adaptive activation. (a) Activated pixels from the single-branch regression. (b - e) Activated pixels for nose, left shoulder, left knee, and left ankle from the multi-branch regression (our approach) at the center pixel for each person. One can see that the proposed approach is able to activate the pixels around the keypoint. The illustrations are obtained using the backbone HRNet-W32.

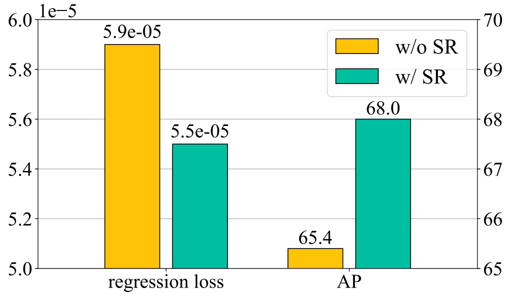

Figure 5. Separate regression improves the regression quality and thus the performance. Using separate regression, the regression loss on COCO train set is reduced from $5.9e$-5 to $5.5e$-5, and the AP score on COCO validation set is increased from 65.4 to 68.0. SR = separate regression. The results are obtained using the backbone HRNet-W32.

input the three kinds of features to a scoring net, consisting of two fully-connected layers (each followed by a ReLU layer), and a linear prediction layer, which aims to learn the OKS score for the corresponding predicted pose with the real OKS as the target on the training set.

3.4. Discussions

Separate regression, group convolution and complexities. In the multi-branch structure, we divide the channel maps into K non-overlapping partitions, and feed each partition into each branch, for learning disentangled representations. This process resembles group convolution. The difference lies in: group convolution usually increases the capacity of the whole representation through reducing the redundancy and increasing the width within the computation and parameter budget, while our approach does not change the width and aims to learn rich representations focusing on each keypoint.

Let's look at an example with HRNet-W32 as the backbone. The standard process in HRNet-W32 concatenates the channels obtained from 4 resolutions and feeds the concatenated channels to a 1×1 convolution, outputting 256 channels. When applied to our disentangled regressors, we modify the 1×1 convolution to output 255 ($= 17 \times 15$) channels, so that each partition has 15 channels. This modification does not increase the width. The parameter complexity and the computation complexity for the regression head are reduced, and in particular the overall computation complexity is significantly reduced. The detailed numbers are given in Table 1.

Table 1. Comparing the parameter and computation complexities between disentangled keypoint regression (DEKR), baseline regression (baseline), baseline + adaptive activation (+ AA), baseline + separation regression (+ SR). Head: only the regression head is counted. Overall: the whole network is counted. The statistical results are from the backbone HRNet-W32.

Method	Head		Overall	
	#param. (M)	GFLOPs	#param. (M)	GFLOPs
baseline	1.31	21.48	30.65	63.28
+ AA	1.34	21.94	30.68	63.73
+ SR	0.19	3.14	29.53	44.93
DEKR	0.22	3.59	29.56	45.39

Separate group regression. It is noticed that the salient regions and the activated pixels for some keypoints might have some overlapping. For example, the salient regions of the five keypoints in the head are overlapped, and three keypoints in the arms have similar characteristics. We investigate the performance if grouping some keypoints into a single branch instead of letting each branch handle one single keypoint. We consider two grouping schemes. (1) Five keypoints in the head use a single branch, and there are totally 13 branches. (2) Five keypoints in the head use a single branch, the three keypoints in left arm (right arm, left leg, right leg) use a single branch. There are totally 5 branches. Empirical results show that separate group regression performs worse than separate regression, e.g., the AP score for five branches decreases by 0.4 on COCO validation with the backbone HRNet-W32.

4. Experiments

4.1. Setting

Dataset. We evaluate the performance on the COCO keypoint detection task [36]. The train2017 set includes 57K images and 150K person instances annotated with 17 keypoints, the val2017 set contains 5K images, and the test-dev2017 set consists of 20K images. We train the models on the train2017 set and report the results on the val2017 and test-dev2017 sets.

Training set construction. The training sets consist of keypoint and center heatmaps, and offset maps.

Groundtruth keypoint and center heatmaps: The

Table 2. Comparisons on the COCO validation set. AE: Associative Embedding [40].

Method	Input size	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{M}	AR^{L}
single-scale testing									
CenterNet-DLA [78]	512	58.9	–	–	–	–	–	–	–
CenterNet-HG [78]	512	64.0	–	–	–	–	–	–	–
PifPaf [31]	–	67.4	–	–	–	–	–	–	–
HGG [27]	512	60.4	83.0	66.2	–	–	64.8	–	–
PersonLab [47]	601	54.1	76.4	57.7	40.6	73.3	57.7	43.5	77.4
PersonLab [47]	1401	66.5	86.2	71.9	62.3	73.2	70.7	65.6	77.9
HrHRNet-W32 + AE [11]	512	67.1	86.2	73.0	–	–	–	61.5	76.1
HrHRNet-W48 + AE [11]	640	69.9	87.2	76.1	–	–	–	65.4	76.4
Our approach (HRNet-W32)	512	68.0	86.7	74.5	62.1	77.7	73.0	66.2	82.7
Our approach (HRNet-W48)	640	71.0	88.3	77.4	66.7	78.5	76.0	70.6	84.0
multi-scale testing									
HGG [27]	512	68.3	86.7	75.8	–	–	72.0	–	–
Point-Set Anchors [67]	640	69.8	88.8	76.3	65.9	76.6	75.6	70.6	83.1
HrHRNet-W32 + AE [11]	512	69.9	87.1	76.0	–	–	–	65.3	77.0
HrHRNet-W48 + AE [11]	640	72.1	88.4	78.2	–	–	–	67.8	78.3
Our approach (HRNet-W32)	512	70.7	87.7	77.1	66.2	77.8	75.9	70.5	83.6
Our approach (HRNet-W48)	640	72.3	88.3	78.6	68.6	78.6	77.7	72.8	84.9

Table 3. GFLOPs and #parameters of the representative top competitors and our approaches with the backbones: HRNet-W32 (DEKR32) and HRNet-W48 (DEKR48). AE-HG = associative embedding-Hourglass.

	AE-HG	PersonLab	HrHRNet	DEKR32	DEKR48
Input size	512	1401	640	512	640
#param. (M)	227.8	68.7	63.8	29.6	65.7
GFLOPs	206.9	405.5	154.3	45.4	141.5

groundtruth keypoint heatmaps H^* for each image contains K maps, and each map corresponds to one keypoint type. We build them as done in [40]: assigning a heat value using the Gaussian function centered at a point around each groundtruth keypoint. The center heatmap is similarly constructed and described in the following.

Groundtruth offset maps: The groundtruth offset maps O^* for each image are constructed from all the poses $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_N\}$. We use the nth pose $\mathcal{P}_n$ as an example and others are the same. We compute the center position $\bar{\mathbf{p}}_n = \frac{1}{K}\sum_{k=1}^{K}\mathbf{p}_{nk}$ and the offsets $\mathcal{T}_n = \{\mathbf{p}_{n1} - \bar{\mathbf{p}}_n, \mathbf{p}_{n2} - \bar{\mathbf{p}}_n, \cdots, \mathbf{p}_{nK} - \bar{\mathbf{p}}_n\}$ as the groundtruth offsets for the pixel corresponding to the center position. We use an expansion scheme to augment the center point to the center region: $\{\mathbf{m}_n^1, \mathbf{m}_n^2, \cdots, \mathbf{m}_n^M\}$, which are the central positions around the center point $\bar{\mathbf{p}}_n$ with the radius 4, and accordingly update the offsets. The positions not lying in the region have no offset values.

Each central position $\mathbf{m}_n^m$ has a confidence value c_n^m indicating how confident it is the center and computed using the way forming the groundtruth center heatmap $\mathbf{C}^*$[2]. The positions not lying in the region have zero heat value.

[2] In case that one position belongs to two or more central regions, we choose only one central region whose center is the closest to that position.

Evaluation metric. We follow the standard evaluation metric[3] and use OKS-based metrics for COCO pose estimation. We report average precision and average recall scores with different thresholds and different object sizes: AP, AP^{50}, AP^{75}, AP^{M}, AP^{L}, AR, AR^{M}, and AR^{L}.

Training. The data augmentation follows [40] and includes random rotation ($[-30°, 30°]$), random scale ($[0.75, 1.5]$) and random translation ($[-40, 40]$). We conduct image cropping to 512×512 for HRNet-W32 or 640×640 for HRNet-W48 with random flipping as training samples.

We use the Adam optimizer [29]. The base learning rate is set as $1\text{e}{-3}$, and is dropped to $1\text{e}{-4}$ and $1\text{e}{-5}$ at the 90th and 120th epochs, respectively. The training process is terminated within 140 epochs.

Testing. We resize the short side of the images to $512/640$ and keep the aspect ratio between height and width, and compute the heatmap and pose positions by averaging the heatmaps and pixel-wise keypoint regressions of the original and flipped images. Following [40], we adopt three scales $0.5, 1$ and 2 in multi-scale testing. We average the three heatmaps over three scales and collect the regressed results from the three scales as the candidates.

4.2. Results

COCO Validation. Table 2 shows the comparisons of our method and other state-of-the-art methods. Table 3 presents the parameter and computation complexities for our approach and the representative top competitors, such as AE-Hourglass [40], PersonLab [47], and HrHRNet [11].

Our approach, using HRNet-W32 as the backbone, achieves 68.0 AP score. Compared to the methods with similar GFLOPs, CenterNet-DLA [78] and PersonLab [47]

[3] http://cocodataset.org/#keypoints-eval

Table 4. Comparisons on the COCO test-dev set. * means using refinement. AE: Associative Embedding.

Method	Input size	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	AR^{M}	AR^{L}
single-scale testing									
OpenPose* [7]	–	61.8	84.9	67.5	57.1	68.2	66.5	–	–
AE [40]	512	56.6	81.8	61.8	49.8	67.0	–	–	–
CenterNet-DLA [78]	512	57.9	84.7	63.1	52.5	67.4	–	–	–
CenterNet-HG [78]	512	63.0	86.8	69.6	58.9	70.4	–	–	–
MDN$_3$ [63]	–	62.9	85.1	69.4	58.8	71.4	–	–	–
PifPaf [31]	–	66.7	–	–	62.4	72.9	–	–	–
SPM* [44]	–	66.9	88.5	72.9	62.6	73.1	–	–	–
PersonLab [47]	1401	66.5	88.0	72.6	62.4	72.3	71.0	66.1	77.7
HrHRNet-W48 + AE [11]	640	68.4	88.2	75.1	64.4	74.2	–	–	–
Our approach (HRNet-W32)	512	67.3	87.9	74.1	61.5	76.1	72.4	65.4	81.9
Our approach (HRNet-W48)	640	70.0	89.4	77.3	65.7	76.9	75.4	69.7	83.2
multi-scale testing									
AE [40]	512	63.0	85.7	68.9	58.0	70.4	–	–	–
AE* [40]	512	65.5	86.8	72.3	60.6	72.6	70.2	64.6	78.1
DirectPose [61]	800	64.8	87.8	71.1	60.4	71.5	–	–	–
SimplePose [33]	512	68.1	–	–	66.8	70.5	72.1	–	–
HGG [27]	512	67.6	85.1	73.7	62.7	74.6	71.3	–	–
PersonLab [47]	1401	68.7	89.0	75.4	64.1	75.5	75.4	69.7	83.0
Point-Set Anchors [67]	640	68.7	89.9	76.3	64.8	75.3	74.8	69.6	82.1
HrHRNet-W48 + AE [11]	640	70.5	89.3	77.2	66.6	75.8	–	–	–
Our approach (HRNet-W32)	512	69.8	89.0	76.6	65.2	76.5	75.1	69.5	82.8
Our approach (HRNet-W48)	640	71.0	89.2	78.0	67.1	76.9	76.7	71.5	83.9

(with the input size 601), our approach achieves over 9.0 improvement. In comparison to CenterNet-HG [78] whose model size is far larger than HRNet-W32, our gain is 4.0. Our baseline result 61.9 (Table 5) is lower than CenterNet-HG that adopts post-processing to match the predictions to the closest keypoints identified from keypoint heatmaps. This implies that our gain comes from our methodology.

Our approach benefits from large input size and large model size. Our approach, with HRNet-W48 as the backbone and the input size 640, obtains the best performance 71.0 and 3.0 gain over HRNet-W32. Compared with state-of-the-art methods, our approach gets 7.0 gain over CenterNet-HG, 4.5 gain over PersonLab (the input size 1401), 3.6 gain over PifPaf [31] whose GFLOPs are more than twice as many as ours, and 1.1 gain over HrHRNet-W48 [11] that uses higher resolution representations.

Following [40, 47], we report the results with multi-scale testing. This brings about 2.7 gain for HRNet-W32, 1.3 gain for HRNet-W48.

COCO test-dev. The results of our approach and other state-of-the-art methods on the test-dev dataset are presented in Table 4. Our approach with HRNet-W32 as the backbone achieves 67.3 AP score, and significantly outperforms the methods with the similar model size. Our approach with HRNet-W48 as the backbone gets the best performance 70.0, leading to 3.5 gain over PersonLab, 3.3 gain over PifPaf [31], and 1.6 gain over HrHRNet [11].

With multi-scale testing, our approach with HRNet-W32 achieves 69.8, even better than PersonLab with a larger model size. Our approach with HRNet-W48 achieves 71.0 AP score, much better than associative embedding [40], 2.3 gain over PersonLab, and 0.5 gain over HrHRNet [11].

Table 5. Ablation study in terms of the AP score, and four types of errors. Adaptive activation (AA) gets 3.5 AP gain over the baseline. Separate regression (SR) further gets 2.6 AP gain. Adaptive activation and separate regression mainly reduce the two localization errors, Jitter and Miss, by 4.6 and 1.5. The results are from COCO validation with the backbone HRNet-W32.

AA	SR	AP	Jitter	Miss	Inversion	Swap
		61.9	16.4	7.6	3.3	1.0
	✓	63.6	15.2	7.2	3.3	1.0
✓		65.4	13.5	6.7	3.1	1.1
✓	✓	68.0	11.8	6.1	3.0	1.1

4.3. Empirical Analysis

Ablation study. We study the effects of the two components: adaptive activation (AA) and separate regression (SR). We use the backbone HRNet-W32 as an example. The observations are consistent for HRNet-W48.

The ablation study results are presented in Table 5. We can observe: (1) adaptive activation (AA) achieves the gain 3.5 over the regression baseline (61.9). (2) separate regression (SR) further improves the AP score by 2.6. (3) separate regression w/o adaptive activation gets 1.7 AP gain. The whole gain is 6.1.

We further analyze how each component contributes to the performance improvement by using the coco-analyze

Table 6. Match regression to the closest keypoints detected from the keypoint heatmaps. Matching does not improve the single-scale (ss) testing performance, and helps multi-scale (ms) testing. Direct regression may need a proper multi-scale testing scheme, which leaves as our future work. D-32 = DEKR with HRNet-W32. D-48 = DEKR with HRNet-W48.

	D-32 (ss)	D-48 (ss)	D-32 (ms)	D-48 (ms)
COCO Val	$68.0_{-0.0}$	$71.0_{-0.0}$	$71.0_{\uparrow 0.3}$	$72.8_{\uparrow 0.5}$
COCO Test	$67.3_{-0.0}$	$70.1_{\uparrow 0.1}$	$70.2_{\uparrow 0.4}$	$71.4_{\uparrow 0.4}$
CrowdPose	$65.5_{\downarrow 0.2}$	$67.0_{\downarrow 0.3}$	$67.5_{\uparrow 0.5}$	$68.3_{\uparrow 0.3}$

tool [53]. Four error types are studied: (i) *Jitter* error: small localization error; (ii) *Miss* error: large localization error; (iii) *Inversion* error: confusion between keypoints within an instance. (iv) *Swap* error: confusion between keypoints of different instances. The detailed definitions are in [53].

Table 5 shows the errors of the four types for four schemes. The two components, adaptive activation (AA) and separate regression (SR), mainly influence the two localization errors, *Jitter* and *Miss*. Adaptive activation reduces the *Jitter* error and the *Miss* error by 2.9 and 0.9, respectively. Separate regression further reduces the two errors by 1.7 and 0.6. The other two errors are almost not changed. This indicates that the proposed two components indeed improve the localization quality.

Comparison with grouping detected keypoints. It is reported in HigherHRNet [11] that associative embedding [40] with HRNet-W32 achieves an AP score 64.4 on COCO validation. The regression baseline using the same backbone HRNet-W32 gets an lower AP score 61.9 (Table 5). The proposed two components lead to an AP score 68.0, higher than associative embedding + HRNet-W32.

Matching regression to the closest keypoint detection. The CenterNet approach [78] performs a post-processing step to refine the regressed keypoint positions by absorbing the regressed keypoint to the closest keypoint among the keypoints identified from the keypoint heatmaps.

We tried this absorbing scheme. The results are presented in Table 6. We can see that the absorbing scheme does not improve the performance in the single-scale testing case. The reason might be that the keypoint localization quality of our approach is very close to that of keypoint identification from the heatmap. In the multi-scale testing case, the absorbing scheme improves the results. The reason is that keypoint position regression is conducted separately for each scale and the absorbing scheme makes the regression results benefit from the heatmap improved from multiple scales. Our current focus is not on multi-scale testing whose practical value is not as high as single-scale testing. We leave finding a better multi-scale testing scheme as our future work.

Table 7. Comparisons on the CrowdPose test set.

Method	Input size	AP	AP^{50}	AP^{75}	AP^{E}	AP^{M}	AP^{H}
single-scale testing							
OpenPose [7]	–	–	–	–	62.7	48.7	32.3
HrHRNet-W48 [11]	640	65.9	86.4	70.6	73.3	66.5	57.9
Ours (HRNet-W32)	512	65.7	85.7	70.4	73.0	66.4	57.5
Ours (HRNet-W48)	640	67.3	86.4	72.2	74.6	68.1	58.7
multi-scale testing							
HrHRNet-W48 [11]	640	67.6	87.4	72.6	75.8	68.1	58.9
Ours (HRNet-W32)	512	67.0	85.4	72.4	75.5	68.0	56.9
Ours (HRNet-W48)	640	68.0	85.5	73.4	76.6	68.8	58.4

4.4. CrowdPose

Dataset. We evaluate our approach on the CrowdPose [34] dataset that is more challenging and includes many crowded scenes. The train set contains $10K$ images, the val set includes $2K$ images and the test set consists of $20K$ images. We train our models on the CrowdPose train and val sets and report the results on the test set as done in [11].

Evaluation metric. The standard average precision based on OKS which is the same as COCO is adopted as the evaluation metrics. The CrowdPose dataset is split into three crowding levels: easy, medium, hard. We report the following metrics: AP, AP^{50}, AP^{75}, as well as AP^{E}, AP^{M} and AP^{H} for easy, medium and hard images.

Training and testing. The train and test methods follow those for COCO except the training epochs. We use the Adam optimizer [29]. The base learning rate is set as $1\text{e}{-3}$, and is dropped to $1\text{e}{-4}$ and $1\text{e}{-5}$ at the 200th and 260th epochs, respectively. The training process is terminated within 300 epochs.

Test set results. The results of our approach and other state-of-the-art methods on the test set are showed in Table 7. Our approach with HRNet-W48 as the backbone achieves 67.3 AP and is better than HrHRNet-W48 (65.9) that is a keypoint detection and grouping approach with a backbone that is designed for improving heatmaps. With multi-scale testing, our approach with HRNet-W48 achieves 68.0 AP score and by a further matching process (see Table 6) the performance is improved, leading to 0.7 gain over HrHRNet-W48 [11].

5. Conclusions

The proposed direct regression approach DEKR improves the keypoint localization quality and achieves state-of-the-art bottom-up pose estimation results. The success stems from that we disentangle the representations for regressing different keypoints so that each representation focuses on the corresponding keypoint region. We believe that the idea of regression by focusing and disentangled keypoint regression can benefit some other methods, such as CornetNet [32] and CenterNet [16] for object detection.

References

[1] Bruno Artacho and Andreas E. Savakis. Unipose: Unified human pose estimation in single images and videos. In *CVPR*, pages 7033–7042, 2020. 2

[2] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *ICCV*, pages 2830–2838, 2015. 2

[3] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. 3

[4] Yanrui Bin, Xuan Cao, Xinya Chen, Yanhao Ge, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Changxin Gao, and Nong Sang. Adversarial semantic data augmentation for human pose estimation. In *ECCV*, pages 606–622, 2020. 2

[5] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, pages 717–732, 2016. 2

[6] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *ECCV*, pages 455–472, 2020. 2

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 1302–1310, 2017. 1, 2, 7, 8

[8] João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *CVPR*, pages 4733–4742, 2016. 2

[9] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *ICCV*, pages 1221–1230, 2017. 2

[10] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018. 2

[11] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020. 1, 6, 7, 8

[12] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *CVPR*, pages 4715–4723, 2016. 2

[13] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *CVPR*, pages 5669–5678, 2017. 2

[14] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 3

[15] Emily L. Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *NeurIPS*, pages 4414–4423, 2017. 3

[16] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6568–6577, 2019. 8

[17] Xiaochuan Fan, Kang Zheng, Yuewei Lin, and Song Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *CVPR*, pages 1347–1355, 2015. 2

[18] Haoshu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: regional multi-person pose estimation. In *ICCV*, pages 2353–2362, 2017. 2

[19] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. Learning to refine human pose estimation. In *CVPR*, pages 205–214, 2018. 2

[20] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, pages 728–743, 2016. 2

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 2

[22] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *CVPR*, pages 5699–5708, 2020. 2

[23] Shaoli Huang, Mingming Gong, and Dacheng Tao. A coarse-fine network for keypoint localization. In *ICCV*, pages 3047–3056, 2017. 2

[24] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, pages 34–50, 2016. 2

[25] Umar Iqbal and Juergen Gall. Multi-person pose estimation with local joint-to-person associations. In *ECCV*, pages 627–642, 2016. 2

[26] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015. 2, 3

[27] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *ECCV*, pages 718–734, 2020. 1, 3, 6, 7

[28] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, pages 731–746, 2018. 2

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 6, 8

[30] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*. 2

[31] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, pages 11977–11986, 2019. 1, 3, 6, 7

[32] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018. 8

[33] Jia Li, Wen Su, and Zengfu Wang. Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation. In *AAAI*, pages 11354–11361, 2020. 7

[34] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*, 2019. 2, 8

[35] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *ECCV*, pages 246–260, 2016. 2

[36] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 5

[37] Wentao Liu, Jie Chen, Cheng Li, Chen Qian, Xiao Chu, and Xiaolin Hu. A cascaded inception of inception network with attention modulated feature fusion for human pose estimation. In *AAAI*, pages 7170–7177, 2018. 2

[38] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, pages 10955–10964, 2019. 3

[39] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In *CVPR*, pages 7773–7781, 2019. 2

[40] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, pages 2274–2284, 2017. 1, 3, 6, 7, 8

[41] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016. 2

[42] Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Pose partition networks for multi-person pose estimation. In *ECCV*, 2018. 3

[43] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, pages 519–534, 2018. 2

[44] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *ICCV*, 2019. 3, 7

[45] Xuecheng Nie, Jiashi Feng, Yiming Zuo, and Shuicheng Yan. Human pose estimation with parsing induced learner. In *CVPR*, 2018. 2

[46] Utku Ozbulak. Pytorch cnn visualizations. https://github.com/utkuozbulak/pytorch-cnn-visualizations, 2019. 1

[47] 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 *ECCV*, pages 282–299, 2018. 3, 6, 7

[48] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, pages 3711–3719, 2017. 2

[49] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S. Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, 2018. 2

[50] Leonid Pishchulin, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *CVPR*, pages 588–595, 2013. 2

[51] 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 *CVPR*, pages 4929–4937, 2016. 2

[52] Lingteng Qiu, Xuanye Zhang, Yanran Li, Guanbin Li, Xiaojun Wu, Zixiang Xiong, Xiaoguang Han, and Shuguang Cui. Peeking into occluded joints: A novel framework for crowd pose estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, pages 488–504, 2020. 2

[53] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, pages 369–378, 2017. 8

[54] Taiki Sekii. Pose proposal networks. In *ECCV*, 2018. 2

[55] Kai Su, Dongdong Yu, Zhenqi Xu, Xin Geng, and Changhu Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *CVPR*, pages 5674–5682, 2019. 2

[56] Ke Sun, Cuiling Lan, Junliang Xing, Wenjun Zeng, Dong Liu, and Jingdong Wang. Human pose estimation using global and local normalization. In *ICCV*, pages 5600–5608, 2017. 2

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

[58] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 536–553, 2018. 2

[59] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *CVPR*, pages 1107–1116, 2019. 3

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

[61] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. In *CoRR*, 2019. 7

[62] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 2

[63] Ali Varamesh and Tinne Tuytelaars. Mixture dense regression for object detection and human pose estimation. In *CVPR*, 2020. 3, 7

[64] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 3

[65] Jian Wang, Xiang Long, Yuan Gao, Errui Ding, and Shilei Wen. Graph-pcnn: Two stage human pose estimation with graph pose refinement. In *ECCV*, pages 492–508, 2020. 2

[66] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[67] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance

segmentation and pose estimation. In *ECCV*, pages 527–544, 2020. 3, 6, 7

[68] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016. 2

[69] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pages 472–487, 2018. 2

[70] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *ICCV*, pages 1290–1299, 2017. 2

[71] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, pages 3073–3082, 2016. 2

[72] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011. 2

[73] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, pages 7091–7100, 2020. 2

[74] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *CVPR*, pages 3517–3526, 2019. 2

[75] Hong Zhang, Hao Ouyang, Shu Liu, Xiaojuan Qi, Xiaoyong Shen, Ruigang Yang, and Jiaya Jia. Human pose estimation with spatial contextual information. In *CoRR*, 2019. 2

[76] Ziyuan Zhang, Luan Tran, Xi Yin, Yousef Atoum, Xiaoming Liu, Jian Wan, and Nanxin Wang. Gait recognition via disentangled representation learning. In *CVPR*, pages 4710–4719, 2019. 3

[77] Lu Zhou, Yingying Chen, Yunze Gao, Jinqiao Wang, and Hanqing Lu. Occlusion-aware siamese network for human pose estimation. In *ECCV*, pages 396–412, 2020. 2

[78] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *CoRR*, 2019. 1, 3, 4, 6, 7, 8

[79] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3VAE: self-supervised sequential VAE for representation disentanglement and data generation. In *CVPR*, pages 6537–6546, 2020. 3