

The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation

Junjie Huang, Zheng Zhu, Feng Guo, Guan Huang, and Dalong Du

Abstract—Being a fundamental component in training and inference, **data processing** has not been systematically considered in human pose estimation community, to the best of our knowledge. In this paper, we focus on this problem and find that the devil of human pose estimation evolution is in the **biased data processing**. Specifically, by investigating the standard data processing in state-of-the-art approaches mainly including **coordinate system transformation** and **keypoint format transformation** (i.e., **encoding** and **decoding**), we find that the results obtained by **common flipping strategy** are **unaligned with the original ones in inference**. Moreover, there is a **statistical error** in some keypoint format transformation methods. Two problems couple together, significantly degrade the pose estimation performance and thus lay a trap for the research community. This trap has given bone to many suboptimal remedies, which are always unreported, confusing but influential. By causing failure in reproduction and unfair in comparison, the unreported remedies seriously impedes the technological development. To tackle this dilemma from the source, we propose **Unbiased Data Processing (UDP)** consist of two technique aspect for the two aforementioned problems respectively (i.e., **unbiased coordinate system transformation** and **unbiased keypoint format transformation**). Base on UDP, we wipe out the trap by giving out a deep insight of the existing biased data processing pipeline, whose origin, effects and some confusing remedies are thoroughly studied. Besides, as a model-agnostic approach and a superior solution, UDP successfully pushes the performance boundary of human pose estimation. For example on COCO *test-dev* set, UDP promotes top-down method HRNet-W32-256 \times 192 by 1.7 AP (73.5 to 75.2) for free and promotes bottom-up methods HRNet-W32-512 \times 512 by 2.7 AP with an acceleration of 6.1 times. The HRNet-W48-384 \times 288 equipped with UDP achieves 76.5 AP and sets a new state-of-the-art for human pose estimation. As a meaningful milestone for pursuing high performance human pose estimation, UDP has been the key base of the winner in 2020 COCO Keypoint Detection Challenge. The code is public available for reference.

Index Terms—Human Pose Estimation, Keypoint Detection, Data Processing.

1 INTRODUCTION

2D Human pose estimation has been extensively studied in computer vision literature and serves many complicated downstream visual understanding tasks such as 3D human pose estimation [1], [2], [3], [4], [5], [6], human phasing [7], [8], health care [9], [10], [11], video surveillance [12], [13], [14], [15] and action recognition [3], [16], [17], [18]. In this paper, we pay attention to the **data processing** aspect, considering it as a fundamental component. All visual recognition tasks are born with data processing, and in general share data processing methodology with each other like **data augmentation and transformation between different coordinate systems**. However, when compared with other tasks like classification [19], object detection [20] and semantic segmentation [21], [22], the performance of human pose estimation algorithms is much more sensitive to the methods used in data processing on account of the evaluation principle. In the evaluation of human pose estimation, the metrics are calculated based on the positional offset between ground truth annotations and predicted results [20], [23], where small disturbance caused by data processing will affect the performance of pose estimators by a large margin.

Although it is of significant, to the best of our knowledge, data processing has not been systematically considered in human pose estimation community. When this topic is addressed, we find

J. Huang, F. Guo, G. Huang and D. Du are with XForwardAI Technology Co.,Ltd, Beijing, China. E-mail: junjie.huang@ieee.org, {feng.guo, guan.huang, dalong.du}@xforwardai.com

Z. Zhu is with Tsinghua University, Beijing, China. E-mail: zhengzhu@ieee.org

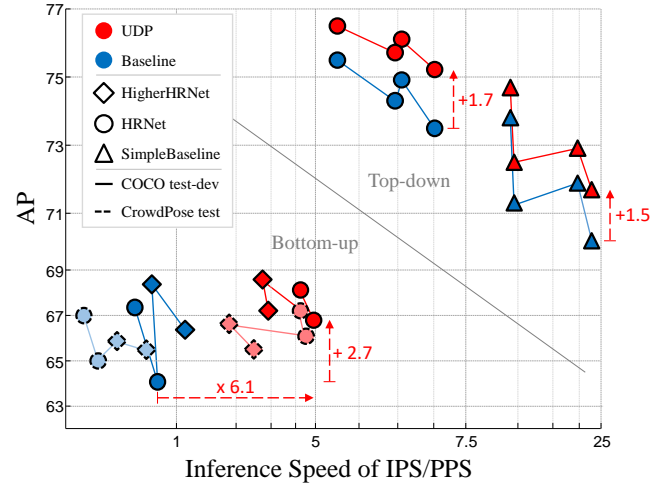


Fig. 1: The improvement of performance on COCO *test-dev* set and CrowdPose *test* set when the proposed Unbiased Data Processing (UDP) is applied to the state-of-the-art methods. At no cost, UDP improves the APs of top-down methods SimpleBaseline [24] and HRNet [25] by a considerable margin. In bottom-up paradigm, UDP offers both accuracy improvement and inference acceleration to HRNet [25] and HigherHRNet [26].

that the widely used data processing pipelines in most state-of-the-art human pose estimation processing systems [24], [25], [26], [27], [28], [29] are defective. The chief causes are two common problems: i)

When flipping testing strategy is adopted, the results from flipped image are unaligned with those from the origin image. The bias derives from utilizing *pixel* for measuring the size of images when performing coordinate system transformation in resizing operation. ii) Defective keypoint format transformation (i.e., encoding and decoding) methods would lead to extra precision degradation. The two problems accumulatively degrade the performance of human pose estimators, lay a trap for the research community and subsequently has given born to many suboptimal remedies. The empirical remedies are always unreported but with huge impact on the performance like direct compensation in post processing [24], [25], [26], [27], [28], [29], while the others are reported but at tremendous cost of latency like using higher network output resolution in HigherHRNet [26]. It is worth noting that, by causing failure in reproduction and unfair in comparison, the unreported remedies will obstruct the development of the human pose estimation technologies.

In this paper, we offer a reasonable and free access to thoroughly solving the two aforementioned problems by proposing Unbiased Data Processing (UDP) system. Corresponding to the two aforementioned problems, UDP consists of two technical aspects: the unbiased coordinate system transformations and the unbiased keypoint format transformations. Aiming at the unbiased coordinate system transformations, we firstly propose to follow the principle of defining and analyzing this problem in continuous space. Then the concept of coordinate system transformation is defined based on this principle and the targets of unbiased in this sub problem are introduced. Subsequently, the coordinate system transformations in different elementary operations (e.g., cropping, resizing, rotating and flipping) are formally designed, which finally compose the common coordinate system transformations used in training and testing process. With mathematical reasoning, we verify the unbiased property of the designed coordinate system transformation pipeline, and subsequently offer a deep insight of the existing biased coordinate system transformation pipeline, whose origin, effects and some confusing remedies are thoroughly studied. Analogously, the concept of unbiased keypoint format transformation is proposed, two unbiased keypoint format transformation methods are introduced and a typical biased example is analyzed thoroughly. As a result with UDP, the aforementioned trap can be remove and a higher as well as more reliable baseline can be achieved.

To showcase the effectiveness of the proposed method, we perform comprehensive experiments on the COCO Keypoint Detection benchmarks [20]. As a model-agnostic approach and a superior solution, UDP successfully pushes the performance boundary of the human pose estimation problem as illustrated in Figure 1. Specifically, UDP boosts the performance of the methods in top-down paradigm without any extra latency. For example, UDP promotes the SimpleBaseline [24] by 1.5 AP (70.2 to 71.7) and 1.0 AP (71.9 to 72.9) within ResNet50-256 \times 192 and ResNet152-256 \times 192 configurations, respectively. For HRNet [25] within W32-256 \times 192 and W48-256 \times 192 configurations, UDP obtains gains by 1.7 AP (73.5 to 75.2) and 1.4 AP (74.3 to 75.7), respectively. The HRNet-W48-384 \times 288 equipped with UDP achieves 76.5 AP (1.0 improvement) and sets a new state-of-the-art for top-down human pose estimation. Besides, in bottom-up paradigm, UDP simultaneously offers both accuracy improvement and latency reducing on the baselines. For HRNet-W32-512 \times 512 configuration in HigherHRNet [26], UDP promotes its performance by 2.7 AP, and at the same time, offers

an acceleration by 6.1 times. For HigherHRNet-W32-512 \times 512 configuration, the promotion and acceleration are +0.8 AP and 2.6 times respectively. In addition, we also perform experiments on extra dataset CrowdPose [30] to verify the generalization ability of UDP among different data distributions. Experimental results show that the performance of UDP in this dataset is in line with that on COCO dataset. Finally to verify the statement in methodology analysis, we measure the contribution of each element in UDP and the effect of the existing remedies in relative works with exhaustive ablation study. Based on the experiment results, we call for attention on the data processing aspect when designing or evaluating the future works. The code is public available for reference.

This paper is built upon our conference paper [31] and significantly extended in several aspects. First, we rearrange the methodology section for methodical stating, and explain it with more specific background introduction and more detailed mathematical reasoning. Second, we extend the coverage of UDP by applying it to methods in bottom-up paradigm and make great discovery and promotion on state-of-the-art method HigherHRNet [26]. Third, we use extra dataset CrowdPose [30] to verify the generalization ability of UDP. In COCO and LVIS 2020 competitions, UDP serves as the baseline for the winner UDP++ [32], which marks this work as a meaningful milestone for pursuing high performance human pose estimation.

2 RELATED WORK

In recent years, research community has witnessed a significant advance from single person [23], [33], [34], [35], [36], [37], [38], [39], [40], [41] to multi-person pose estimation [20], [25], [26], [27], [42], [43], [44], [45], [46], [47], [48], [49], [50], where the latter can be generally categorized into bottom-up [26], [44], [45], [46], [49], [51] and top-down [24], [25], [27], [28], [47], [50], [52], [53] approaches.

Bottom-up methods start by detecting identity-free joints for all the persons in an input image and then group them into person instances. In this paradigm, both cost and efficient are considered, both the identity-free joint detection and grouping strategy are the main concerns. OpenPose [46] builds a model that contains two branches to predict keypoint heatmaps and pairwise relationships (part affinity fields) between them, where the latter acts as the main cue in grouping process. MultiPoseNet [54] simultaneously achieves human detection and pose estimation, and proposes PRN to group the keypoints by the bounding box of each people. Aiming at resolving the human pose estimation problem in crowd sense, Li et al. [30] design a new model by combining joint-candidate single person pose estimation and global maximum joints association. Simultaneously, a new dataset named CrowdPose is collected specific for performance evaluation in crowd senses. Newell et al. [49] use one network for both heatmap prediction and embedding study. Grouping is done by utilizing association embedding, which assigns each keypoint with a tag and groups keypoints based on the L2 distance between tag vectors. As a follower, Chen et al. [26] replace the hourglass style networks in [49] with the proposed HigherHRNet. By using higher output resolution, HigherHRNet improves the precision of the predictions by a large margin.

. <https://github.com/HuangJunJie2017/UDP-Pose>

. <https://cocodataset.org/workshop/coco-lvis-eccv-2020.html>

Top-down methods achieve multi-person pose estimation by the two-stages process, including obtaining person bounding boxes through a person detector like Faster R-CNN [55] and predicting keypoint locations within these boxes. As single person pose estimation is performed with fixed scale patches, most state-of-the-art performances on multi-person popular benchmarks COCO [20] are achieved by top-down methods [27], [28], [31]. Existing works with this paradigm pay more attention to the designing of network structure. Chen et al. [56] propose Structure-aware Convolutional Network trained with Generative Adversarial Networks for human pose structure exploiting. Following ShuffleNet [57] and SENet [58], Su et al. [59] propose Channel Shuffle Module (CSM) and Spatial, Channel-wise Attention Residual Bottleneck (SCARB) specific for human pose estimation problem. CPN [27] and MSPN [28] are the leading methods on COCO keypoint challenge, adopting cascade network to refine the keypoints prediction. SimpleBaseline [24] adds a few deconvolutional layers to enlarge the resolution of output features. Thought simple, it has a competitive performance among existing works. HRNet [25] maintains high-resolution representations through the whole process, achieving state-of-the-art performance on public datasets. Mask R-CNN [53] builds an end-to-end framework and achieves a good balance between performance and inference speed.

Data processing in human pose estimation mainly includes *coordinate system transformation* and *keypoint format transformation*. **Coordinate system transformation** means transforming the data (i.e., keypoint coordinates and image matrixes) between different coordinate systems when some operations are conducted like cropping, resizing, rotating and flipping. During this process, most state-of-the-art methods [24], [25], [26], [27], [28] use *pixel* to measure the size of images when performing resizing operation, leading to unaligned results when using flipping strategy in inference. This bias degrades the accuracy by a large margin, lays a trap for research community and has given bone to some suboptimal remedies. The remedies are all empirical and always unreported. For example, without any explanation, SimpleBaseline [24], HRNet [25] and Darkpose [29] empirically shift the result from flipped image by 1 pixel in network output coordinate system to suppress the predicting error. CPN [27] and MSPN [28] achieve similar effect by shifting the average result by 2 pixels in network input coordinate system. HigherHRNet [26] proposes to use higher network output resolution and conducts the experiment with some unreported compensation for large superiority on the baseline. These remedies are effective and appealing, but being the recipe for disaster as they hinder the development of technology by causing failure in reproduction and unfair in comparison. In this paper, we propose unbiased coordinate system transformation to thoroughly solve this problem, which will not only boost the performance of the existing methods but also provide a more reliable baseline for future works. **Keypoint format transformation** (i.e., encoding and decoding) commonly denotes the transformation between joint coordinates and heatmaps, which is firstly proposed in [38] and has been widely used in state-of-the-art methods [24], [25], [26], [27], [28], [53]. In training process, it encodes the annotated keypoint coordinate into a heatmap with Gaussian distribution. And in testing process, it decodes the network predicted heatmap back into keypoint coordinate. This pipeline shows superior performance when compared with directly predicting the keypoint coordinates [60], but is still imperfect on account of its defective design and inherent precision degradation. The combined classification and regression format based encoding-

decoding paradigm in [47] provides an mathematically error-free entrance to further promote the prediction accuracy. Analogously, Darkpose [29] achieve unbiased keypoint format transformation by proposing a distribution-aware decoding method to match the encoding method use in [26]. In this paper, we will introduce these **two unbiased keypoint format transformation paradigm**, verify their unbiased property and show their superiority on the baseline.

- 1, unbiased coordinate system transformation
- 2, unbiased keypoint format transformation

3 UNBIASED DATA PROCESSING FOR HUMAN POSE ESTIMATION

In human pose estimation, data processing involves the transformation between different coordinate system and the transformation between different keypoint format. In the following, we will give details introduction of our unbiased data processing method in these **two aspects** respectively (i.e., *unbiased coordinate system transformation* and *unbiased keypoint format transformation*).

3.1 Unbiased Coordinate System Transformation

As it is new to this topic and quite ambiguous to community, for clarified and reasonable statement, the concept of **unbiased coordinate system transformation** is constructed from the base. We firstly propose **the unified definition of data in continuous space**. Then based on this definition, the concept of *coordinate system transformation* and *the targets of unbiased* are introduced. We design the coordinate system transformation in some elementary operations (i.e., cropping, resizing, rotating and flipping) before we construct the common composite transformations between the coordinate systems involved in human pose estimation problem (i.e., *source image coordinate system*, *network input coordinate system* and *network output coordinate system*). Subsequently, we verify the unbiased properties of the designed coordinate system transformation pipeline with mathematical reasoning. And at last to showcase how the defective coordinate system affect the research community, some biased data processing methods are analyzed, and the theory behind some reported techniques and unreported tricks used in state-of-the-arts are thoroughly studied.

3.1.1 An Unified Definition of Data in Continuous Space.

The image matrixes and the target keypoint coordinates are the main data involved in human pose estimation problem. The images are stored and processed in a discrete format, but the keypoint coordinates are defined, processed and evaluated in continuous spaces. To avoid precision degradation in the coordinate system transformation pipeline, an unified paradigm is required for uniformly analyzing and dealing with different data in the coordinate system transformation problems.

To this end, we assume that there is a continuous image plane and consider each image matrix as a discrete sampling result on it, where each pixel in an image matrix is a specific sample point. Formally, in line with the definition of target keypoint coordinates in COCO dataset [20], we define the coordinate system $O-XY$ as illustrated in Figure 2 to describe the continuous image planes. The origin of the coordinate system is located at the most top-left pixel, the $O-X$ direction is from left to right and the $O-Y$ direction is from top to down. Besides, the distance between adjacent pixels is assumed to be equivalent and is defined as the unit length of the coordinate system. Then we have an image matrix as a sampling result of the image plane \mathbf{I} , which is denoted as $\{\mathbf{I}(\mathbf{p}) = (r, g, b) | \mathbf{p} = (x, y), x \in \{0, 1, 2, \dots, w\}, y \in \{0, 1, 2, \dots, h\}\}$. w and

像素是离散的
坐标是连续的
量化是有误差的

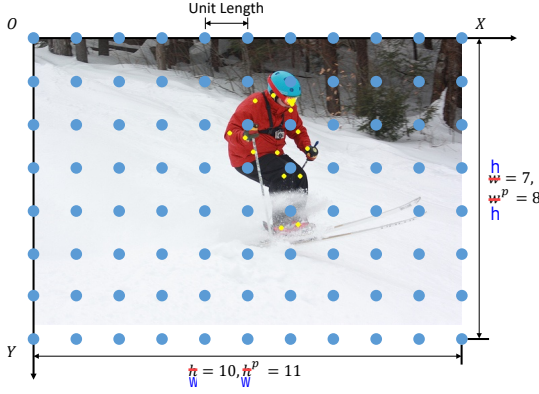


Fig. 2: Illustration of analyzing the coordinate system transformation problem in continuous space. $O-XY$ denotes the coordinate system. An image matrix (the set of blue points) is regarded as a sampling result of the continuous image plane.

h are the width and height of the image counted in unit lengths. And a set of target keypoints are also defined in the same image plane and denoted as $\{\mathbf{k} = (x, y)\}$.

It is worth noting that the size of the sample points defined here is infinitely small and the size of the images' semantically meaningful area is calculated with the unit length. As a result, the image size (i.e., w for width and h for height) we discussed following is different from the resolution of the image matrix, which is widely used for defining the image size in common sense. Formally, the relationship between them is as follow:

$$\begin{aligned} w &= w^p - 1 \\ h &= h^p - 1 \end{aligned} \quad (1)$$

where w^p and h^p are the width and height of the image matrix counted in pixels. We use superscript p to discriminate the variables counted in pixel from those measured in unit length.

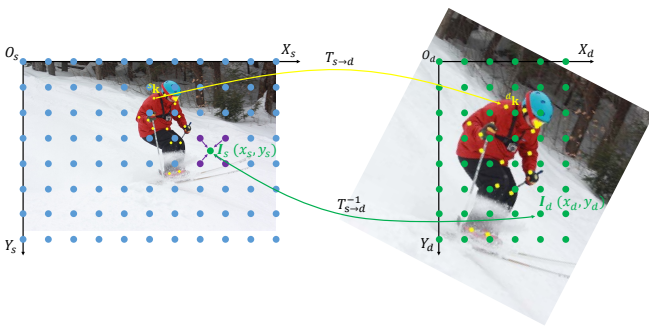


Fig. 3: The illustration of the coordinate system transformation in human pose estimation problem.

3.1.2 The Concept of Coordinate System Transformation.

The *coordinate system transformation* in human pose estimation can be generally formulated as the data description transformation from the source coordinate system into the destination coordinate system. As illustrated in Figure 3, we label the source coordinate system with subscript s as $O_s - X_s Y_s$ and the target coordinate

system as $O_d - X_d Y_d$. Then the transformation of keypoint coordinates can be formulated as:

$$\mathbf{k}_d = T_{s \rightarrow d} \mathbf{k}_s \quad (2)$$

where $T_{s \rightarrow d}$ is the coordinate system transformation matrix from the source coordinate system to the destination coordinate system. And the transformation of the contents in image matrix can be formulated as:

$$\mathbf{I}_d(\mathbf{p}_d) = \mathbf{I}_s(T_{s \rightarrow d}^{-1} \mathbf{p}_d) \quad (3)$$

变换前后，像素点的rgb值保持不变
image matrices

where $T_{s \rightarrow d}^{-1}$ is the inverse of $T_{s \rightarrow d}$. Equation 3 means that we make the image constant semantically aligned with the annotated keypoints in the destination coordinate system by setting the color of position \mathbf{p}_d the same as that in the source image at position $T_{s \rightarrow d}^{-1} \mathbf{p}_d$. The results of backtracking $T_{s \rightarrow d}^{-1} \mathbf{p}_d$ are usually not integers, and thus, $\mathbf{I}_s(T_{s \rightarrow d}^{-1} \mathbf{p}_d)$ should be calculated by bilinear interpolation with the valid surrounding points (i.e., the purple points in Figure 3). As we only have a sampling result (i.e., the image matrix) of the image plane, interpolation is the optimal way to reduce the precision degradation in image transformation, but can not thoroughly remedy it. Thus, as the precision degradation of interpolation is irreversible and cumulative, we have a principle that the less interpolation done in the data processing pipeline is the better in designing coordinate system transformation pipelines.

3.1.3 The Targets of Unbias.

Unbias is a target in coordinate system transformation designing, which contains **two aspect**: One is to keep the **semantical alignment** after performing transformations. Semantical alignment means that the positional relativeness between different data (i.e., images and keypoint positions) is unchanged (e.g., the annotated position of nose in destination space is still exactly located upon the nose in the image described in destination space). This is guaranteed by keeping the transformation matrix the same in both Equation 2 and Equation 3.

Another aspect is to make the predicting result exactly aligned with the ground truth under the assumption that the network has a perfect learning ability. In other words, we hope the network's learning ability to be the unique source of precision degradation, and there are no defects in the design of the coordinate transformation pipeline will cause precision degradation. In the following, we will detail our unbiased coordinate system transformation pipeline and prove its unbiased property.

3.1.4 Coordinate System Transformation in Elementary Operations.

Coordinate system transformations in human pose estimation derives from some elementary operations like cropping, resizing, rotating and flipping.

Cropping, as illustrated in Figure 4, is conducted according to a specific Region of Interest (ROI) defined in the source coordinate system $ROI = (bx_s, by_s, bw_s, bh_s)$, where (bx_s, by_s) denotes its center position and (bw_s, bh_s) denotes its width and height. The destination coordinate system can be obtained by moving the origin of the source coordinate system to the upper left corner of the ROI. Thus, the transformation matrix should be designed as:

$$T_{crop}(ROI) = \begin{bmatrix} 1 & 0 & -bx_s + 0.5bw_s \\ 0 & 1 & -by_s + 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

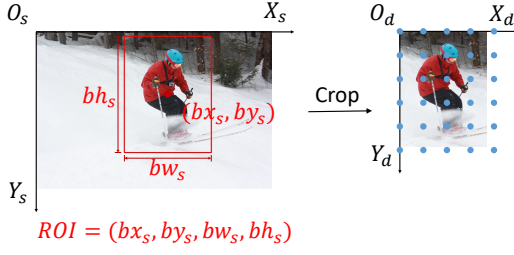


Fig. 4: The coordinate system transformation in cropping operation.

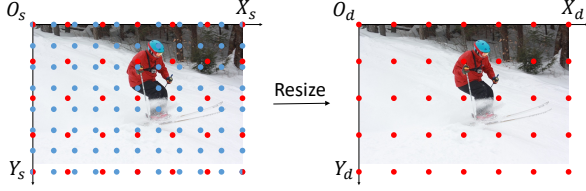


Fig. 5: The coordinate system transformation in resizing operation.

Resizing, as illustrated in Figure 5, changes the sampling strategy only and keep the semantic constant of the image the same as the source. We make the four corner sample point semantically align with the source four corner sample point and let the other sample points evenly distributed among the area dividing by the four corners. Thus the only thing that changes is the unit length of the coordinate system and the transformation matrix should be designed as:

$$T_{resize}(w_s, h_s, w_d, h_d) = \begin{bmatrix} \frac{w_d}{w_s} & 0 & 0 \\ 0 & \frac{h_d}{h_s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

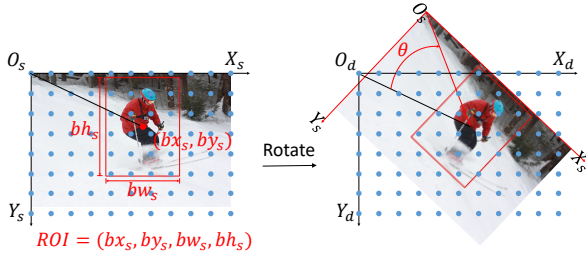


Fig. 6: The coordinate system transformation in rotating operation.

Rotating, as illustrated in Figure 6, is conducted according to a rotation center which is always set as the center of a specific ROI instead of the origin of the coordinate system. This design aims at keeping the center position of ROI unchanged (i.e., $(bx_s, by_s) = (bx_d, by_d)$). For example, the ROI refers to the bounding boxes of human instances in top-down paradigm and the whole image in bottom-up paradigm. So, the transformation matrix should be designed as the combination of three elementary transformations:

$$\begin{aligned} T_{rot}(\theta, ROI) &= T_{d2 \rightarrow d1} T_{d1 \rightarrow d2} T_{s \rightarrow d1} \\ &= \begin{bmatrix} 1 & 0 & bx_s \\ 0 & -1 & by_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -bx_s \\ 0 & -1 & by_s \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & -\sin \theta & -bx_s \cos \theta + by_s \sin \theta + bx_s \\ \sin \theta & \cos \theta & -bx_s \sin \theta - by_s \cos \theta + by_s \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (6)$$

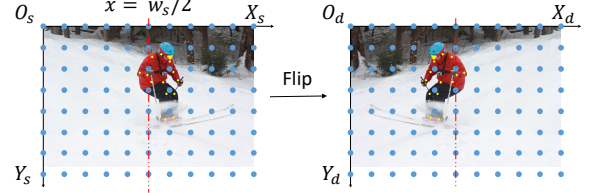


Fig. 7: The coordinate system transformation in flipping operation.

Flipping, as illustrated in Figure 7, generally takes $x = w_s/2$ as the mirror and horizontally exchanges the images' content. So, the transformation matrix should be designed as:

$$T_{flip}(w_s) = \begin{bmatrix} -1 & 0 & w_s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

3.1.5 Common Coordinate System Transformation

In human pose estimation as illustrated in Figure 8, there are **three coordinate systems** are involved: source image coordinate systems denoted as $O_s-X_sY_s$ with subscript s corresponding to the source image with a size of (w_s, h_s) , network input coordinate systems denoted as $O_i-X_iY_i$ with subscript i corresponding to the network input with a size of (w_i, h_i) , and network output coordinate systems denoted as $O_o-X_oY_o$ with subscript o corresponding to the network output with a size of (w_o, h_o) .

During the training process, the data is firstly transformed from the source image coordinate systems into the network input coordinate systems according to a specific ROI (bx_s, by_s, bw_s, bh_s) and a rotation angle θ . Some elementary operations are conducted orderly:

$$\begin{aligned} T_{flip}(w_i) &= \begin{bmatrix} -1 & 0 & w_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ T_{rot}(\theta, (0.5w_i, 0.5h_i, w_i, h_i)) &= \begin{bmatrix} \cos \theta & -\sin \theta & -0.5w_i \cos \theta + 0.5h_i \sin \theta + 0.5w_i \\ \sin \theta & \cos \theta & -0.5w_i \sin \theta - 0.5h_i \cos \theta + 0.5h_i \\ 0 & 0 & 1 \end{bmatrix} \\ T_{resize}(bw_s, bh_s, w_i, h_i) &= \begin{bmatrix} \frac{w_i}{bw_s} & 0 & 0 \\ 0 & \frac{h_i}{bh_s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ T_{crop}(bx_s, by_s, bw_s, bh_s) &= \begin{bmatrix} 1 & 0 & -bx_s + 0.5bw_s \\ 0 & 1 & -by_s + 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (8)$$

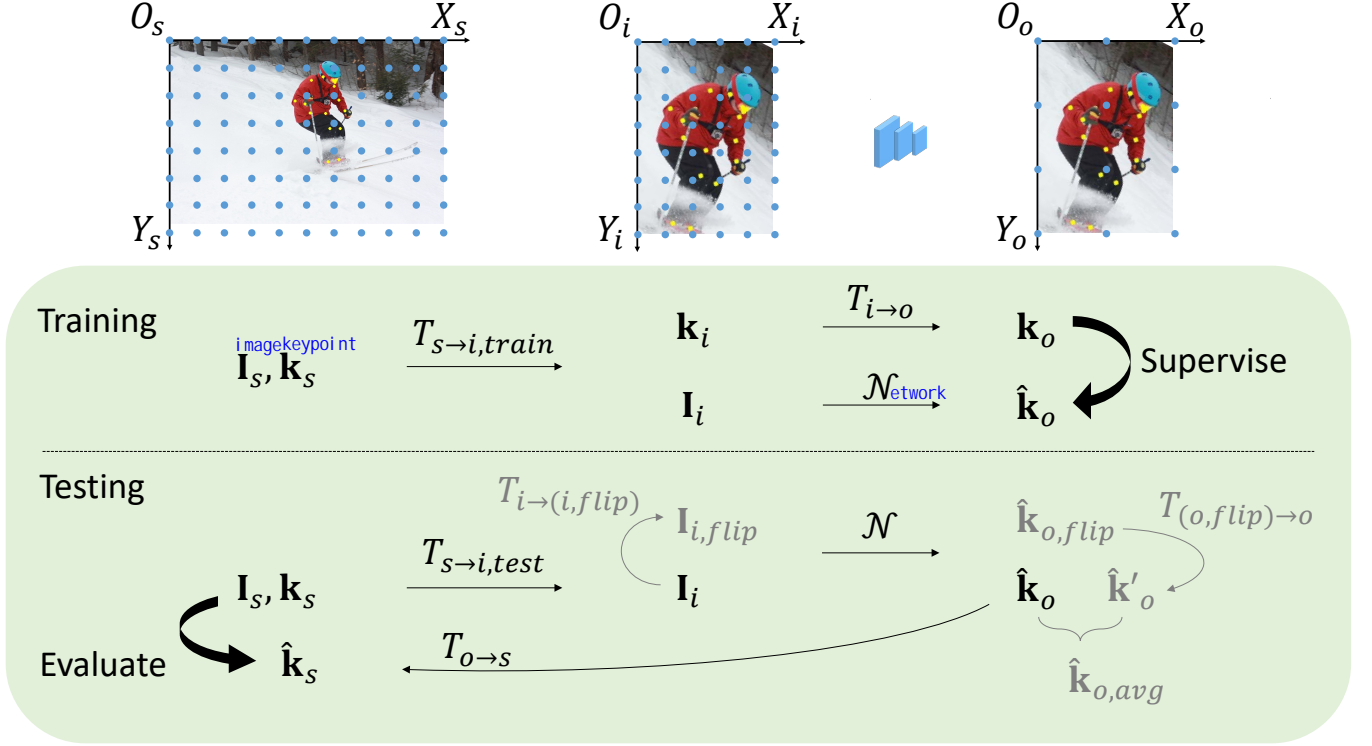


Fig. 8: The illustration of the common coordinate system transformation in human pose estimation problem. Three coordinate system are involved: source image coordinate system O_s - X_sY_s , network input coordinate system O_i - X_iY_i and network output coordinate system O_o - X_oY_o .

Then we have the combined transformation:

$$\begin{aligned} \mathbf{k}_i &= T_{s \rightarrow i, \text{train}} \mathbf{k}_s \\ \mathbf{I}_i(\mathbf{p}_i) &= \mathbf{I}_s(T_{s \rightarrow i, \text{train}}^{-1} \mathbf{p}_i) \\ T_{s \rightarrow i, \text{train}} &= T_{\text{flip}} T_{\text{rot}} T_{\text{resize}} T_{\text{crop}} \end{aligned} \quad (9)$$

Equation 9 integrates not only the necessary transformations like cropping and resizing, but also the optional augmentations (i.e., T_{flipping} for random flipping, T_{rotating} for random rotating, T_{cropping} for half body and random cropping.) used in human pose estimator training. Cropping and Resizing are necessary, while flipping and rotating are optional. The image matrixes in network input space are set as the network input and we have the inference results in the network output space:

$$\hat{\mathbf{k}}_o = \mathcal{N}(\mathbf{I}_i) \quad (10)$$

where \mathcal{N} denotes the networks. The annotation is simultaneously transformed from the network input space into the network output space by a simple resizing operation:

$$\begin{aligned} \mathbf{k}_o &= T_{i \rightarrow o} \mathbf{k}_i \\ T_{i \rightarrow o} &= T_{\text{resize}} \\ T_{\text{resize}}(w_i, h_i, w_o, h_o) &= \begin{bmatrix} \frac{w_o}{w_i} & 0 & 0 \\ 0 & \frac{h_o}{h_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (11)$$

And \mathbf{k}_o in the network output space serves as the supervision:

$$\text{Loss} = \|\hat{\mathbf{k}}_o - \mathbf{k}_o\| \quad (12)$$

The networks are optimized in the training process and as an ideal result, we have:

$$\begin{aligned} \text{Loss} &= \|\hat{\mathbf{k}}_o - \mathbf{k}_o\| = 0 \\ \mathcal{N}(\mathbf{I}_i) &= \hat{\mathbf{k}}_o = \mathbf{k}_o = T_{i \rightarrow o} \mathbf{k}_i \end{aligned} \quad (13)$$

which means that the network learns not only the reflection from image matrixes \mathbf{I}_i to keypoint positions \mathbf{k}_i , but also the reflection of the transformation $T_{i \rightarrow o}$ defined in Equation 11.

In testing process, only the image matrixes are transformed from the source image coordinate systems into the network input coordinate system with the necessary elementary transformations, which should be in line with those in the training process:

$$\begin{aligned} \mathbf{I}_i(\mathbf{p}_i) &= \mathbf{I}_s(T_{s \rightarrow i, \text{test}}^{-1} \mathbf{p}_i) \\ T_{s \rightarrow i, \text{test}} &= T_{\text{resize}} T_{\text{crop}} \\ T_{\text{resize}}(bw_s, bh_s, w_i, h_i) &= \begin{bmatrix} \frac{w_i}{bw_s} & 0 & 0 \\ 0 & \frac{h_i}{bh_s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ T_{\text{crop}}(bx_s, by_s, bw_s, bh_s) &= \begin{bmatrix} 1 & 0 & -bx_s + 0.5bw_s \\ 0 & 1 & -by_s + 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (14)$$

Then the network outputs in Equation 10 are transformed back to

the source image coordinate systems by inverse transformations:

$$\begin{aligned}
\hat{\mathbf{k}}_s &= T_{o \rightarrow s} \hat{\mathbf{k}}_o \\
T_{o \rightarrow s} &= T_{crop} T_{resize} \\
T_{resize}(w_o, h_o, bw_s, bh_s) &= \begin{bmatrix} \frac{bw_s}{w_o} & 0 & 0 \\ 0 & \frac{bh_s}{h_o} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
T_{crop}(0.5w_s - bx_s + 0.5bw_s, & \\ 0.5y_s - by_s + 0.5bh_s, w_s, h_s) & \\ = \begin{bmatrix} 1 & 0 & bx_s - 0.5bw_s \\ 0 & 1 & by_s - 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} &
\end{aligned} \quad (15)$$

With Equation 13 as assumption and taking Equation 11, Equation 14, Equation 15 into consideration, we have the following identical relation:

$$\begin{aligned}
\hat{\mathbf{k}}_s &= T_{o \rightarrow s} \hat{\mathbf{k}}_o \\
&= T_{o \rightarrow s} T_{i \rightarrow o} \mathbf{k}_i \\
&= T_{o \rightarrow s} T_{i \rightarrow o} T_{s \rightarrow i, test} \mathbf{k}_s \\
&= \begin{bmatrix} 1 & 0 & bx_s - 0.5bw_s \\ 0 & 1 & by_s - 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{bw_s}{w_o} & 0 & 0 \\ 0 & \frac{bh_s}{h_o} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_o}{w_i} & 0 & 0 \\ 0 & \frac{h_o}{h_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{w_i}{bw_s} & 0 & 0 \\ 0 & \frac{h_i}{bh_s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -bx_s + 0.5bw_s \\ 0 & 1 & -by_s + 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_s \\
&= \mathbf{k}_s
\end{aligned} \quad (16)$$

This inference prove that the result in the source image space is exactly equal to the ground truth, which means that the data transformation pipeline designed above is unbiased and no systematic error would be involved.

When flipping ensemble is used in testing process, the flipped image is obtained by performing flipping transformation in the network input space:

$$\begin{aligned}
\mathbf{I}_{i, flip}(\mathbf{p}_{i, flip}) &= \mathbf{I}_i(T_{i \rightarrow (i, flip)}^{-1} \mathbf{p}_{i, flip}) \\
T_{i \rightarrow (i, flip)} &= T_{flip}(w_i) = \begin{bmatrix} -1 & 0 & w_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \quad (17)$$

Then we have the network prediction $\hat{\mathbf{k}}_{o, flip} = \mathcal{N}(\mathbf{I}_{i, flip})$ which is subsequently flipped back in the network output space:

$$\begin{aligned}
\hat{\mathbf{k}}'_o &= T_{(o, flip) \rightarrow o} \hat{\mathbf{k}}_{o, flip} \\
T_{(o, flip) \rightarrow o} &= T_{flip}(w_o) = \begin{bmatrix} -1 & 0 & w_o \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \quad (18)$$

with Equation 13 as assumption and taking Equation 11, Equation 17 and Equation 18 into consideration, we have the following

identical relation:

$$\begin{aligned}
\hat{\mathbf{k}}'_o &= T_{(o, flip) \rightarrow o} \hat{\mathbf{k}}_{o, flip} \\
&= T_{(o, flip) \rightarrow o} T_{(i, flip) \rightarrow (o, flip)} \mathbf{k}_{i, flip} \\
&= T_{(o, flip) \rightarrow o} T_{(i, flip) \rightarrow (o, flip)} T_{i \rightarrow (i, flip)} \mathbf{k}_i \\
&= T_{flip}(w_o) T_{resize}(w_i, h_i, w_o, h_o) T_{flip}(w_i) \mathbf{k}_i \\
&= \begin{bmatrix} -1 & 0 & w_o \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_o}{w_i} & 0 & 0 \\ 0 & \frac{h_o}{h_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & w_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_i \\
&= \begin{bmatrix} \frac{w_o}{w_i} & 0 & 0 \\ 0 & \frac{h_o}{h_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_i \\
&= T_{i \rightarrow o} \mathbf{k}_i \\
&= \hat{\mathbf{k}}_o
\end{aligned} \quad (19)$$

This inference prove that, in the network output space, the results from flipped images are aligned with those from the origin images. By taking Equation 16 into consideration, the results from flipped images in the source image space are also aligned with the ground truths and no systematic error would be involved. The establish of Equation 16 and Equation 19 guarantees the unbiased property in the coordinate system transformation pipeline. They will be used as the guideline for checking biased coordinate system transformation pipelines in the following subsection.

3.1.6 Diagnosis of the Biased Coordinate System Transformation

In most state-of-the-arts [24], [25], [26], [27], [28], the bias problem in coordinate system transformation pipeline derives from using resolution (w_s^p, h_s^p) counted in pixels instead of size (w_s, h_s) measured in unit length when performing resizing transformation. As a consequence, it changes Equation 16 and Equation 19 into:

$$\begin{aligned}
\hat{\mathbf{k}}_s &= T_{o \rightarrow s} \hat{\mathbf{k}}_o \\
&= T_{o \rightarrow s} T_{i \rightarrow o} \mathbf{k}_i \\
&= T_{o \rightarrow s} T_{i \rightarrow o} T_{s \rightarrow i, test} \mathbf{k}_s \\
&= \begin{bmatrix} 1 & 0 & bx_s - 0.5bw_s \\ 0 & 1 & by_s - 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{bw_s}{w_o^p} & 0 & 0 \\ 0 & \frac{bh_s}{h_o^p} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_o^p}{w_i^p} & 0 & 0 \\ 0 & \frac{h_o^p}{h_i^p} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{w_i^p}{bw_s} & 0 & 0 \\ 0 & \frac{h_i^p}{bh_s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -bx_s + 0.5bw_s \\ 0 & 1 & -by_s + 0.5bh_s \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_s \\
&= \mathbf{k}_s
\end{aligned} \quad (20)$$

$$\begin{aligned}
\hat{\mathbf{k}}'_o &= T_{(o,flip) \rightarrow o} \hat{\mathbf{k}}_{o,flip} \\
&= T_{(o,flip) \rightarrow o} T_{(i,flip) \rightarrow (o,flip)} \mathbf{k}_{i,flip} \\
&= T_{(o,flip) \rightarrow o} T_{(i,flip) \rightarrow (o,flip)} T_{i \rightarrow (i,flip)} \mathbf{k}_i \\
&= T_{flip}(w_o) T_{resize}(w_i^p, h_i^p, w_o^p, h_o^p) T_{flip}(w_i) \mathbf{k}_i \\
&= \begin{bmatrix} -1 & 0 & w_o \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_o^p}{w_i^p} & 0 & 0 \\ 0 & \frac{h_o^p}{h_i^p} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & w_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_i \\
&= \begin{bmatrix} \frac{w_o^p}{w_i^p} & 0 & \frac{w_o^p}{w_i^p} - 1 \\ 0 & \frac{h_o^p}{h_i^p} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_i \\
&= \begin{bmatrix} 1 & 0 & \frac{w_o^p}{w_i^p} - 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_o^p}{w_i^p} & 0 & 0 \\ 0 & \frac{h_o^p}{h_i^p} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{k}_i \\
&= \begin{bmatrix} 1 & 0 & \frac{1-s}{s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o
\end{aligned} \tag{21}$$

where $s = w_i^p/w_o^p$ is the stride factor for describing the size variation of network features. Here, $\hat{\mathbf{k}}_s$ is still equal to \mathbf{k}_s , indicating that the aforementioned modification will not change the unbiased property in coordinate system transformation pipeline $T_{o \rightarrow s} T_{i \rightarrow o} T_{s \rightarrow i, test}$ and should have no effect on the precision of the predicted results. However when flipping ensemble is adopted in testing process, $\hat{\mathbf{k}}'_o$ is not exactly aligned with $\hat{\mathbf{k}}_o$ and there is an offset of $\frac{1-s}{s}$ in O_o-X_o direction. Taking $\hat{\mathbf{k}}_o$ as reference, $\frac{1-s}{s}$ is the predicting error of result $\hat{\mathbf{k}}'_o$ in network output space. If we directly average $\hat{\mathbf{k}}'_o$ and $\hat{\mathbf{k}}_o$ as done in most existing works:

$$\hat{\mathbf{k}}_{o,avg} = \frac{\hat{\mathbf{k}}'_o + \hat{\mathbf{k}}_o}{2} = \begin{bmatrix} 1 & 0 & \frac{1-s}{2s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o \tag{22}$$

the final error in O_o-X_o direction is:

$$e(x)_o = |x(\hat{\mathbf{k}}_{o,avg}) - x(\hat{\mathbf{k}}_o)| = \left| \frac{1-s}{2s} \right| = 0.375|_{s=4} \tag{23}$$

where $\hat{\mathbf{k}}_o$ is regarded as ground truth as it has been proved unbiased by Equation 20. The magnitude of this predicting error is so large that the performance will be degraded by a considerable margin. In state-of-the-arts, there are some empirical remedies for this error, which can be classified into two categories: direct compensation or using higher resolution.

As the error $\left| \frac{1-s}{2s} \right|$ has a fixed scale which is determined by the stride factor, direct compensation is effective, being the remedy in most state-of-the-arts top-down methods [24], [25], [27], [28], [29]. For example, SimpleBaseline [24], HRNet [25] and DarkPose [29] empirically shift the result from flipped image by one pixel in O_o-X_o direction before performing the averaging operation to suppress this error:

$$\hat{\mathbf{k}}_{o,avg} = \frac{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}'_o + \hat{\mathbf{k}}_o}{2} = \begin{bmatrix} 1 & 0 & \frac{1}{2s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o \tag{24}$$

In this way, the final error can be reduced to

$$e(x)'_o = \left| \frac{1}{2s} \right| = 0.125|_{s=4} \tag{25}$$

where $e(x)'_o < e(x)_o$ when $s > 2$, which makes sense in most existing top-down methods with a stride factor of 4 [24], [25], [27], [28], [29]. Intuitively, as a result of reasoning, an extra compensation for $e(x)'_o$ in network output space can make the result of existing work more accurate. We will verify this in ablation study.

Besides, when mapping $e(x)'_o$ back to source image coordinate system ($O_s-X_sY_s$) with Equation 15, we have:

$$e(x)'_s = \left| \frac{1}{2s} \times \frac{bw_s}{w_o^p} \right| = \left| \frac{bw_s}{2w_i^p} \right| \tag{26}$$

where bw_s is fixed in inference process. Equation 26 means that higher network input resolution can help suppress the predicted error caused by $e(x)'_s$. In other words, the existing top-down methods benefit more from higher input resolution and suffer more accuracy loss from lower input resolution.

Without shifting one pixel in network output space, we have:

$$e(x)_s = \left| \frac{s-1}{2s} \times \frac{bw_s}{w_o^p} \right| = \left| \frac{bw_s(s-1)}{2w_i^p} \right| \tag{27}$$

which means that both higher input resolution and higher output resolution can help suppress this error. And this contributes the most performance boosting in HigherHRnet [26] who empirically proposes to use higher output resolution to pursue high precision at the cost of tremendous latency in both network inference and post processing. By contrast, unbiased data processing provides a free access to achieve similar performance improvement with a low output resolution. Besides of using higher output resolution, HigherHRNet uses another unreported operation that resizes the network output into a resolution as high as the network input. This operation also coincidentally remedies the error caused by biased coordinate system transformation pipeline and benefit the performance of HigherHRNet structure at the cost of extra latency in post processing. Through ablation study, we will show this operation are gilding the lily when the coordinate system transformation pipeline is unbiased as it will involve extra error and change the distribution of the network output by performing an extra interpolation in resizing operation.

3.2 Unbiased Keypoint Format Transformation

3.2.1 The Concept of Unbiased Keypoint Format Transformation.

As the coordinate of keypoint is not the superior format for convolutional network study, the intuitively more proper format of heatmap is proposed and quickly has been proved effective. The *keypoint fromat transformation* refers to the transformations between keypoint coordinates and heatmaps which is widely used in state-of-the-art methods. In common sense, *encoding* denotes the transformation from coordinate format into heatmap format, while *decoding* denotes the inverse transformation.

$$\begin{aligned}
\mathcal{H} &= \text{Encoding}(\mathbf{k}) \\
\mathbf{k} &= \text{Decoding}(\mathcal{H})
\end{aligned} \tag{28}$$

Target of *Unbiased* in keypoint format transformation designing is to avoid precision degeneration in the encoding and decoding transformation. As a formulated target, we should have:

$$\mathbf{k} = \text{Decoding}(\text{Encoding}(\mathbf{k})) \tag{29}$$

3.2.2 Unbiased Keypoint Format Transformation.

In this subsection, we will introduce two unbiased keypoint format transformation paradigm and simultaneously showcase their unbiased property.

Combined classification and regression format is inspired by the works in object detection [55] where anchors are used to predict bounding boxes, and first proposed in [47]. We give details introduction here with some modifications. In training process, each annotated keypoint $\mathbf{k} = (m, n)$ is encoded through:

$$\begin{aligned} \mathcal{C}(x, y, m, n) &= \begin{cases} 1 & \text{if } (x - m)^2 + (y - n)^2 < r^2 \\ 0 & \text{otherwise} \end{cases} \\ \mathcal{X}(x, y, m, n) &= m - x \\ \mathcal{Y}(x, y, m, n) &= n - y \end{aligned} \quad (30)$$

where \mathcal{C} is the classification heatmap act as the anchor in object detection for preliminarily locate the keypoint. r is a hyper-parameter referring the radius of the area classified as positive. Consist of offset vectors, \mathcal{X} and \mathcal{Y} are the regression heatmap for preserving the residual locating information. Then the loss is designed as:

$$\begin{aligned} Loss &= Loss_{cls} + Loss_{reg} \\ Loss_{cls} &= ||\mathcal{C} - \hat{\mathcal{C}}|| \\ Loss_{reg} &= \mathcal{C} * ||\mathcal{X} - \hat{\mathcal{X}}|| + \mathcal{C} * ||\mathcal{Y} - \hat{\mathcal{Y}}|| \\ \hat{\mathcal{C}}, \hat{\mathcal{X}}, \hat{\mathcal{Y}} &= \mathcal{N}(I) \end{aligned} \quad (31)$$

where \mathcal{C} in $Loss_{reg}$ defines the region of interesting, which means that we only need to learn the offset among the area where the classification label is true. The network is optimized in the training process and as a ideal result, we have:

$$\hat{\mathcal{C}}, \hat{\mathcal{X}}, \hat{\mathcal{Y}} = \mathcal{C}, \mathcal{X}, \mathcal{Y} \quad (32)$$

Then in testing processing, the prediction is decoding by:

$$\begin{aligned} \hat{\mathbf{k}} &= \hat{\mathbf{k}}_h + (\hat{\mathcal{X}}(\hat{\mathbf{k}}_h), \hat{\mathcal{Y}}(\hat{\mathbf{k}}_h)) \\ \hat{\mathbf{k}}_h &= \text{argmax}(\hat{\mathcal{C}}) \end{aligned} \quad (33)$$

where the position of highest response $\hat{\mathbf{k}}_h$ is located first and is subsequently updated by utilizing the predicted offsets. By taking Equation 30 and Equation 32 into consideration, we have:

$$\begin{aligned} \hat{\mathbf{k}} &= \hat{\mathbf{k}}_h + (\hat{\mathcal{X}}(\hat{\mathbf{k}}_h), \hat{\mathcal{Y}}(\hat{\mathbf{k}}_h)) \\ &= (x(\hat{\mathbf{k}}_h), y(\hat{\mathbf{k}}_h)) + (m - x(\hat{\mathbf{k}}_h), n - y(\hat{\mathbf{k}}_h)) \\ &= (m, n) \\ &= \mathbf{k} \end{aligned} \quad (34)$$

which means that no systematic error is involved in the keypoint format transformation pipeline and the unbiased target in Equation 29 is achieved.

Classification format is widely used in most state-of-the-arts, where classification heatmap is used only with a gaussian-like distribution:

$$\mathcal{C}(x, y, m, n) = \exp\left(-\frac{(x - m)^2 + (y - n)^2}{2\delta^2}\right) \quad (35)$$

The loss is designed as:

$$\begin{aligned} Loss &= ||\mathcal{C} - \hat{\mathcal{C}}|| \\ \hat{\mathcal{C}} &= \mathcal{N}(I) \end{aligned} \quad (36)$$

The network is optimized in the training process and as a ideal result, we have:

$$\hat{\mathcal{C}} = \mathcal{C} \quad (37)$$

In testing process, we introduce the decoding method DARK [29] who decoding the classification heatmap into keypoint coordinates by searching the center of the gaussian distribution where the first derivative is equal to zero:

$$\begin{aligned} \hat{\mathbf{k}} &= \hat{\mathbf{k}}_h - \hat{\mathcal{C}}''(\hat{\mathbf{k}}_h)^{-1} \hat{\mathcal{C}}'(\hat{\mathbf{k}}_h) \\ \hat{\mathbf{k}}_h &= \text{argmax}(\hat{\mathcal{C}}) \end{aligned} \quad (38)$$

where $\hat{\mathcal{C}}'$ and $\hat{\mathcal{C}}''$ are the first order derivative and second order derivative (i.e., Hessian) of $\hat{\mathcal{C}}$. According to [29], the precision degradation caused by Taylor series approximation is negligible and $\hat{\mathbf{k}}$ is theoretically close to \mathbf{k} , which matches the purpose of unbiased.

3.2.3 Analysis of Biased Keypoint Format Transformation.

We take the keypoint format transformation method used in SimpleBaseline [24], HRNet [25] and HigherHRNet [26] as the example for studying the effect of biased data form transformation. keypoints are encoded into classification heatmap with gaussian distribution as Equation 35, but are decoded by the a suboptimal method:

$$\begin{aligned} \hat{\mathbf{k}} &= \hat{\mathbf{k}}_h + 0.25 * \text{sign}(\hat{\mathcal{C}}'(\hat{\mathbf{k}}_h)) \\ \hat{\mathbf{k}}_h &= \text{argmax}(\hat{\mathcal{C}}) \\ \text{sign}(x) &= \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases} \end{aligned} \quad (39)$$

According to the encoding in Equation 35, we have

$$\begin{aligned} \text{argmax}(\hat{\mathcal{C}}) &= \begin{cases} \text{Floor}(m) & \text{if } m - \text{Floor}(m) < 0.5 \\ \text{Ceil}(m) & \text{otherwise} \end{cases} \\ \text{sign}(\hat{\mathcal{C}}'(\hat{\mathbf{k}}_h)) &= \begin{cases} 1 & \text{if } m - \text{Floor}(m) < 0.5 \\ -1 & \text{otherwise} \end{cases} \end{aligned} \quad (40)$$

As an example, predicting coordinate in $O - X$ direction has the distribution of:

$$\hat{m} = \begin{cases} \text{Floor}(m) + 0.25 & \text{if } m - \text{Floor}(m) < 0.5 \\ \text{Ceil}(m) - 0.25 & \text{otherwise} \end{cases} \quad (41)$$

With the assumption that \mathbf{k} is uniformly distributed in the image plane (i.e., both $m - \text{Floor}(m)$ and $n - \text{Floor}(n)$ are uniformly distributed in interval $[0, 1)$), the expected error in each direction is $E(|m - \hat{m}|) = E(|n - \hat{n}|) = 1/8 = 0.125$ unit length with a variance of $V(|m - \hat{m}|) = V(|n - \hat{n}|) = 1/192 \approx 0.0052$.

When mapping $E(|m - \hat{m}|)$ back to the source image coordinate system ($O_s - X_s Y_s$) with Equation 15, we have:

$$E(|m_s - \hat{m}_s|) = E(|m_o - \hat{m}_o|) \times \frac{bw_s}{w_o} \quad (42)$$

Considering error $E(|m - \hat{m}|)$ and $E(|n - \hat{n}|)$, the methods with biased data form transformation benefit from higher network output resolution. And this also contributes part of the performance boosting in HigherHRnet [26].

TABLE 1: Performance of proposed UDP on COCO *val* set. IPS used in bottom-up paradigm denotes the inference speed of Image Per Second. PPS used in top-down paradigm denotes the inference speed of Person Per Second. † means unreported results in the original paper and trained with official implementation by us.

Method	Backbone	Input size	IPS/PPS	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
Bottom-up methods									
HigherHRNet [26]	HRNet-W32	512 × 512	0.8	64.4	-	-	57.1	75.6	-
+UDP	HRNet-W32	512 × 512	4.9 (×6.1)	67.0 (+2.6)	86.2	72.0	60.7	76.7	71.6
HigherHRNet [26]	HigherHRNet-W32	512 × 512	1.1	67.1	86.2	73.0	61.5	76.1	71.8
+UDP	HigherHRNet-W32	512 × 512	2.9 (×2.6)	67.8 (+0.7)	86.2	72.9	62.2	76.4	72.4
HigherHRNet [26]†	HRNet-W48	640 × 640	0.6	67.9	86.7	74.4	62.5	76.2	73.0
+UDP	HRNet-W48	640 × 640	4.1 (×6.8)	68.9 (+1.0)	87.3	74.9	64.1	76.1	73.5
HigherHRNet [26]	HigherHRNet-W48	640 × 640	0.75	69.9	87.2	76.1	65.4	76.4	-
+UDP	HigherHRNet-W48	640 × 640	2.7 (×3.6)	69.9	87.3	76.2	65.9	76.2	74.4
Bottom-up methods with multi-scale (1×2, ×1, ×0.5) test as in HigherHRNet [26]									
UDP	HRNet-W32	512 × 512	-	70.4	88.2	75.8	65.3	77.6	74.7
HigherHRNet [26]	HigherHRNet-W32	512 × 512	-	69.9	87.1	76.0	65.3	77.0	-
+UDP	HigherHRNet-W32	512 × 512	-	70.2 (+0.3)	88.1	76.2	65.4	77.4	74.5
HigherHRNet [26]†	HRNet-W48	640 × 640	-	71.6	88.6	77.9	67.5	77.8	76.3
+UDP	HRNet-W48	640 × 640	-	71.3 (-0.3)	89.0	77.1	66.9	77.7	75.7
HigherHRNet [26]	HigherHRNet-W48	640 × 640	-	72.1	88.4	78.2	67.8	78.3	-
+UDP	HigherHRNet-W48	640 × 640	-	71.5 (-0.6)	88.3	77.3	67.9	77.2	75.9
Top-down methods									
Hourglass [40]	Hourglass	256 × 192	-	66.9	-	-	-	-	-
CPN [27]	ResNet-50	256 × 192	-	69.4	-	-	-	-	-
CPN [27]	ResNet-50	384 × 288	-	71.6	-	-	-	-	-
MSPN [28]	MSPN	256 × 192	-	75.9	-	-	-	-	-
SimpleBaseline [24]	ResNet-50	256 × 192	23.0	71.3	89.9	78.9	68.3	77.4	76.9
+UDP	ResNet-50	256 × 192	23.0	72.9(+1.6)	90.0	80.2	69.7	79.3	78.2
SimpleBaseline [24]	ResNet-152	256 × 192	11.5	72.9	90.6	80.8	69.9	79.0	78.3
+UDP	ResNet-152	256 × 192	11.5	74.3(+1.4)	90.9	81.6	71.2	80.6	79.6
SimpleBaseline [24]	ResNet-50	384 × 288	20.3	73.2	90.7	79.9	69.4	80.1	78.2
+UDP	ResNet-50	384 × 288	20.3	74.0(+0.8)	90.3	80.0	70.2	81.0	79.0
SimpleBaseline [24]	ResNet-152	384 × 288	11.1	75.3	91.0	82.3	71.9	82.0	80.4
+UDP	ResNet-152	384 × 288	11.1	76.2(+0.9)	90.8	83.0	72.8	82.9	81.2
HRNet [25]	HRNet-W32	256 × 192	6.9	75.6	91.9	83.0	72.2	81.6	80.5
+UDP	HRNet-W32	256 × 192	6.9	76.8(+1.2)	91.9	83.7	73.1	83.3	81.6
HRNet [25]	HRNet-W48	256 × 192	6.3	75.9	91.9	83.5	72.6	82.1	80.9
+UDP	HRNet-W48	256 × 192	6.3	77.2(+1.3)	91.8	83.7	73.8	83.7	82.0
HRNet [25]	HRNet-W32	384 × 288	6.2	76.7	91.9	83.6	73.2	83.2	81.6
+UDP	HRNet-W32	384 × 288	6.2	77.8(+1.1)	91.7	84.5	74.2	84.3	82.4
HRNet [25]	HRNet-W48	384 × 288	5.3	77.1	91.8	83.8	73.5	83.5	81.8
+UDP	HRNet-W48	384 × 288	5.3	77.8(+0.7)	92.0	84.3	74.2	84.5	82.5

3.2.4 Join Analysis of Biased Coordinate System Transformation and Biased Keypoint Format Transformation.

Error ${}^oe(x)' = \frac{1}{2s}$ in Equation 25 has an impact on the decoding result distribution. With a specific stride factor $s = 4$ and considering Equation 21, we have:

$$\begin{aligned}
 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}'_o &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{1-s}{s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o \\
 &= \begin{bmatrix} 1 & 0 & \frac{1}{s} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o \\
 &= \begin{bmatrix} 1 & 0 & 0.25 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{k}}_o
 \end{aligned} \quad (43)$$

As a result, the predicted heatmap from flipped image in $O_o-X_oY_o$ is changed into $\hat{\mathcal{C}}'_o = \mathcal{C}(x, y, m + 0.25, n)$, and the average heatmap distribution is changed into:

$$\begin{aligned}
 \hat{\mathcal{C}}_{o,avg} &= \frac{\mathcal{C}(x, y, m + 0.25, n) + \mathcal{C}(x, y, m, n)}{2} \\
 &\approx \mathcal{C}(x, y, m + 0.125, n)
 \end{aligned} \quad (44)$$

where we use a approximation to simplified the following analysis. Finally, error ${}^oe(x)' = \frac{1}{2s}$ leads to a variation of the result distribution in Equation 39:

$$\hat{m} = \begin{cases} \text{Floor}(m) + 0.25 & \text{if } m - \text{Floor}(m) < 0.375 \\ \text{Ceil}(m) - 0.25 & \text{if } 0.375 \leq m - \text{Floor}(m) < 0.875 \\ \text{Ceil}(m) + 0.25 & \text{otherwise} \end{cases} \quad (45)$$

and the expected error in O_o-X_o direction is enlarged by just 1/32 unit length to $E(|m - \hat{m}|) = 5/32 \approx 0.156$ with a larger variance of $V(|m - \hat{m}|) = 37/3072 \approx 0.012$.

Considering the error ${}^oe(x) = \frac{s-1}{2s}$ in Equation 23, the distribution of decoding result in Equation 39 will change into:

$$\hat{m} = \begin{cases} \text{Floor}(m) - 0.25 & \text{if } m - \text{Floor}(m) < 0.375 \\ \text{Floor}(m) + 0.25 & \text{if } 0.375 \leq m - \text{Floor}(m) < 0.875 \\ \text{Ceil}(m) - 0.25 & \text{otherwise} \end{cases} \quad (46)$$

and the expected error in O_o-X_o direction is enlarged by 1/4 unit length to $E(|m - \hat{m}|) = 3/8 = 0.375$ with a larger variance of $V(|m - \hat{m}|) = 1/48 \approx 0.0208$. Compared with ${}^oe(x) = \frac{s-1}{2s} = 0.375$, the biased decoding method contributes a variance which will have extra negative impact on the final performance. It is worth noting that, the actual errors are more complicated than

that analyzed above, as the approximation in Equation 44 also has an impact on the errors.

4 EXPERIMENTS

4.1 Result on COCO dataset

4.1.1 Implementation Details

For *top-down* paradigm, we take SimpleBaseline [24] and HRNet [25] as baseline and use the official implementation. All training settings are preserved except for the data processing pipeline proposed in this paper. Unbiased keypoint format transformation in combined classification and regression format is used in this paradigm as default with hyper-parameters $r = 0.0625 * w_o^p$ in Equation 30. Classification format is verified in ablation study with hyper-parameters $\delta = 2.0$ in Equation 35. During inference, **HTC** [70] detector is used to detect human instances. With multi-scale test, the 80-class and person AP on COCO *val* set [20] are 52.9 and 65.1, respectively. The results of HRNet [25] and SimpleBaseline [24] on COCO *val* set with this human detection are reproduced for fair comparison. The inference speed is tested on *val* set and measured in Person Per Second (PPS). The hardware environment mainly includes a single RTX 2080ti GPU and an Intel(R) Xeon(R) E5-2630-v4@2.20GHz CPU.

For *bottom-up* paradigm, we take HigherHRNet [26] as baseline and both HRNet and HigherHRNet network structures are exploited. All training settings are preserved except for the data processing pipeline proposed in this paper. During inference, the operation of resizing the network output is removed and the decoding method is replaced with the unbiased one in Equation 38. Testing with single scale and multi-scale (i.e., $[\times 2, \times 1, \times 0.5]$, where $\times 2$ means that the input resolution is enlarged by factor 2 like 512×512 to 1024×1024) are reported respectively. The inference speed is measured in Image Per Second (IPS).

4.1.2 Results of top-down paradigm on the val set.

We report the performance improvement when UDP is applied to SimpleBaseline [24] and HRNet [25] in Table 1. Considering the series of SimpleBaseline, the promotions are +1.6 AP (71.3 to 72.9) for ResNet-50 backbone and +1.4 AP (72.9 to 74.3) for ResNet-152 backbone. For higher network input resolution, the promotions are +0.8 AP and +0.9 AP respectively. For HRNet family, the promotion is +1.2 AP (75.6 to 76.8) for HRNet-w32 backbone and +1.3 AP (75.9 to 77.2) for HRNet-w48 backbone. For higher network input resolution, the promotions are +1.1 AP and +0.7 AP respectively. We summarize some key characteristics of the results: i) improvements are consistent among different backbone types, which indicates that the learning ability of the network has little impact on the precision loss caused by the biased data processing pipeline. This indicates that more powerful network structures proposed in future work would not help solving the bias problem and UDP is the necessary solution. ii) improvements on methods with smaller network input resolution are more than that with larger network input resolution. This is in line with the analysis in methodology that larger network input size can help suppressing the error and models with smaller network input size suffer more precision degression. iii) No extra latency is involved in the proposed method, which means that UDP provides the aforementioned improvement at no cost.

4.1.3 Results of bottom-up paradigm on the val set.

We take the most recent method HigherHRNet [26] as the representative baseline with two network constructions HRNet and HigherHRNet. With biased data processing as reported in [26], HRNet-W32-512 \times 512 configuration only scores 64.4 AP with an inference speed of 0.8 IPS and HigherHRNet-W32-512 \times 512 configuration 67.1 AP with an inference speed of 1.1 IPS. By contrast with UDP, HRNet-W32-512 \times 512 configuration scores 67.0 AP with an inference speed of 4.9 IPS which has 2.6 AP superiority and 6.1 times faster than the baseline. The performance of this configuration is even close to the baseline with HigherHRNet-W32-512 \times 512 configuration, and still 4.5 times faster than it. HigherHRNet-W32-512 \times 512-UDP configuration scores 67.8 AP with an inference speed of 2.9 IPS, which has 0.7 AP superiority and 2.6 times faster than the baseline configuration HigherHRNet-W32-512 \times 512. At no cost, UDP offers both performance boosting and latency reducing. With UDP, we have a more reasonable performance difference between HRNet-W32-512 \times 512 and HigherHRNet-W32-512 \times 512 on COCO *val* set, which is +0.8 AP improvement at the cost of +70% extra latency in inference.

4.1.4 Results on the test-dev set.

Table 2 and Figure 1 report the performance of UDP on COCO *test-dev* set. The results show similar improvement compared with *val* set, indicating the steady generalization property of UDP. Specifically, our approach promotes SimpleBaseline by 1.5 AP (70.2 to 71.7) and 1.0 AP (71.9 to 72.9) within ResNet50-256 \times 192 and ResNet152-256 \times 192 configurations, respectively. For HRNet within W32-256 \times 192 and W48-256 \times 192 configurations, UDP obtains gains by 1.7 AP (73.5 to 75.2) and 1.4 AP (74.3 to 75.7), respectively. The HRNet-W48-384 \times 288 equipped with UDP achieves 76.5 AP and sets a new state-of-the-art for human pose estimation.

4.2 Results on CrowdPose dataset

We utilize the CrowdPose [30] dataset to verify the generalization ability of UDP among different data distributions. HigherHRNet [26] is used as baseline and the experimental configurations are set the same as those in COCO dataset. In line with [26], models are trained on *train* and *val* sets and tested on *test* set. We report the improvement of AP on Table 3. According to the experimental results, UDP not only promotes the accuracy of all configurations, but also speeds up the inference by a large margin. This is in line with that in COCO dataset. The exceptional thing is that, when UDP is applied, HigherHRNet-W32-512 \times 512 configuration (65.6 AP with 2.4 IPS inference speed) and HigherHRNet-W48-640 \times 640 configuration (66.7 AP with 1.8 IPS inference speed) with higher output resolution doesn't show any superiority on HRNet-W32 configuration (66.1 AP with 4.5 IPS inference speed) and HRNet-W48-640 \times 640 configuration (67.2 AP with 4.2 IPS inference speed). This puts doubt on the generalization of the techniques proposed in HigherHRNet [26]. Thus we empirically argue that, by effecting the performance and misguided the researchers, the biased data processing pipeline has a negative effect on the technology development.

4.3 Ablation Study on Top-down Paradigm

In this subsection, we use HRNet-W32 backbone and 256×192 input size to perform ablation study on the techniques involved in

. <https://github.com/leoxiaobin/deep-high-resolution-net.pytorch>
. <https://github.com/HRNet/HigherHRNet-Human-Pose-Estimation>

TABLE 2: The improvement of AP on COCO *test-dev* set when the proposed UDP is applied to state-of-the-art methods. † means unreported results in the original paper and trained with official implementation by us.

Method	Backbone	Input size	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
Bottom-up methods								
AE [49]	Hourglass [40]	512×512	56.6	81.8	61.8	49.8	67.0	-
G-RMI [47]	ResNet-101	353×257	64.9	85.5	71.3	62.3	70.0	69.7
PersonLab [51]	ResNet-152	1401×1401	66.5	88.0	72.6	62.4	72.3	-
PifPaf [61]	-	-	66.7	-	-	-	-	-
HigherHRNet [26]	HRNet-W32	512×512	64.1	86.3	70.4	57.4	73.9	-
+UDP	HRNet-W32	512×512	66.8 (+2.7)	88.2	73.0	61.1	75.0	71.5
HigherHRNet [26]	HigherHRNet-W32	512×512	66.4	87.5	72.8	61.2	74.2	-
+UDP	HigherHRNet-W32	512×512	67.2 (+0.8)	88.1	73.6	62.0	74.3	72.0
HigherHRNet [26]†	HRNet-W48	640×640	67.4	88.6	74.2	62.6	74.3	72.8
+UDP	HRNet-W48	640×640	68.1 (+0.2)	88.3	74.6	63.9	74.1	73.1
HigherHRNet [26]	HigherHRNet-W48	640×640	68.4	88.2	75.1	64.4	74.2	-
+UDP	HigherHRNet-W48	640×640	68.6 (+0.2)	88.2	75.5	65.0	74.0	73.5
Bottom-up methods with multi-scale ($(\times 2, \times 1, \times 0.5)$) test as in HigherHRNet [26]								
UDP	HRNet-W32	512×512	69.3	89.2	76.0	64.8	76.0	74.1
HigherHRNet [26]†	HRNet-W32	512×512	68.8	88.8	75.7	64.4	75.0	73.5
UDP	HigherHRNet-W32	512×512	69.1	89.1	75.8	64.4	75.5	73.8
HigherHRNet [26]†	HRNet-W48	640×640	70.4	89.7	77.4	66.4	75.7	75.2
+UDP	HRNet-W48	640×640	70.3	90.1	76.7	66.6	75.3	75.1
HigherHRNet [26]	HigherHRNet-W48	640×640	70.5	89.3	77.2	66.6	75.8	-
+UDP	HigherHRNet-W48	640×640	70.5	89.4	77.0	66.8	75.4	75.1
Top-down methods								
Mask-RCNN [53]	ResNet-50-FPN [62]	-	63.1	87.3	68.7	57.8	71.4	-
Integral Pose Regression [63]	ResNet-101 [64]	256×256	67.8	88.2	74.8	63.9	74.0	-
SCN [56]	Hourglass [40]	-	70.5	88.0	76.9	66.0	77.0	-
CPN [27]	ResNet-Inception	384×288	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [65]	PyraNet [66]	320×256	72.3	89.2	79.1	68.0	78.6	-
CFN [67]	-	-	72.6	86.1	69.7	78.3	64.1	-
CPN(ensemble) [27]	ResNet-Inception	384×288	73.0	91.7	80.9	69.5	78.1	79.0
Posefix [68]	ResNet-152	384×288	73.6	90.8	81.0	70.3	79.8	79.0
CSANet [69]	ResNet-152	384×288	74.5	91.7	82.1	71.2	80.2	80.7
MSPN [28]	MSPN [28]	384×288	76.1	93.4	83.8	72.3	81.5	81.6
SimpleBaseline [27]	ResNet-50	256×192	70.2	90.9	78.3	67.1	75.9	75.8
+UDP	ResNet-50	256×192	71.7 (+1.5)	91.1	79.6	68.6	77.5	77.2
SimpleBaseline [27]	ResNet-50	384×288	71.3	91.0	78.5	67.3	77.9	76.6
+UDP	ResNet-50	384×288	72.5 (+1.2)	91.1	79.7	68.8	79.1	77.9
SimpleBaseline [27]	ResNet-152	256×192	71.9	91.4	80.1	68.9	77.4	77.5
+UDP	ResNet-152	256×192	72.9 (+1.0)	91.6	80.9	70.0	78.5	78.4
SimpleBaseline [27]	ResNet-152	384×288	73.8	91.7	81.2	70.3	80.0	79.1
+UDP	ResNet-152	384×288	74.7 (+0.9)	91.8	82.1	71.5	80.8	80.0
HRNet [25]	HRNet-W32	256×192	73.5	92.2	82.0	70.4	79.0	79.0
+UDP	HRNet-W32	256×192	75.2 (+1.7)	92.4	82.9	72.0	80.8	80.4
HRNet [25]	HRNet-W32	384×288	74.9	92.5	82.8	71.3	80.9	80.1
+UDP	HRNet-W32	384×288	76.1 (+1.2)	92.5	83.5	72.8	82.0	81.3
HRNet [25]	HRNet-W48	256×192	74.3	92.4	82.6	71.2	79.6	79.7
+UDP	HRNet-W48	256×192	75.7 (+1.4)	92.4	83.3	72.5	81.4	80.9
HRNet [25]	HRNet-W48	384×288	75.5	92.5	83.3	71.9	81.5	80.5
+UDP	HRNet-W48	384×288	76.5 (+1.0)	92.7	84.0	73.0	82.4	81.6

the data processing pipeline. Techniques we study here includes Unbiased Coordinate System Transformation (UCST), Flipping Testing (FT), Shift the Network Output by One Pixel (SNOOP) used in some state-of-the-arts [24], [25], [29], Extra Compensation (EC) proposed in Section 3.1.6 for the residual error left by using SNOOP, Unbiased Keypoint Format Transformation in Combined Classification and Regression Form (UKFT-CCRF), Unbiased Keypoint Format Transformation in Classification Form (UKFT-CF). Experimental settings and the corresponding performance on COCO *val* set are listed in Table 4.

When FT is absent, configuration A and B have similar performances (74.5 AP and 74.4 AP) which is guaranteed by the establish of Equation 16 and Equation 20. This verify the conjecture that using resolution counted in pixels instead of size measured in unit length when performing resizing transformation has no impact on the unbiased property of the data processing pipeline. However, when FT is adopted, the performance of

configuration C doesn't shows any improvement on configuration A, and instead, even drops by 1.2 AP from 74.5 AP to 73.3 AP. This showcase the tremendous negative effect of the error $e(x)_o$ reported in Equation 23. The trap caused by biased coordinate system transformation pipeline is so deep that producing great demand for remedies. By contrast with the proposed UCST, configuration D (75.7 AP) has 1.3 AP improvement on configuration B (74.4 AP). UCST is the prerequisite for performance improving with FT.

By performing an empirical compensation, configuration E with SNOOP scores 75.6 AP, which is close to the result in configuration D with UCST. This means that, by taking the unbiased configuration D as reference, 66.7% of error $e(x)_o$ suppressed by SNOOP has a dominating effect on the performance, and the remainder (i.e., $e(x)'_o$) would have little impact on the performance (i.e., around 0.1 AP, $75.6 \rightarrow 75.7$). We subsequently perform EC in configuration F to verify this. According to the experimental

TABLE 3: The improvement of AP on CrowdPose *test* set when UDP is applied. † means unreported results in the original paper and trained with official implementation by us.

Method	Backbone	Input size	IPS	AP	AP ⁵⁰	AP ⁷⁵	AP ^E	AR ^M	AR ^H
SPPE [30]	ResNet-101	320 × 240	-	66.0	84.2	71.5	75.5	66.3	57.4
HigherHRNet [26]†	HRNet-W32	512 × 512	0.4	65.0	85.9	69.7	72.6	65.4	57.7
+UDP	HRNet-W32	512 × 512	4.5 (×11.3)	66.1 (+1.1)	86.7	70.9	73.5	66.6	58.2
HigherHRNet [26]†	HigherHRNet-W32	512 × 512	0.7	65.5	85.9	70.5	72.8	66.0	57.7
+UDP	HigherHRNet-W32	512 × 512	2.4 (×3.4)	65.6 (+0.1)	86.5	70.5	73.1	66.2	57.5
HigherHRNet [26]†	HRNet-W48	640 × 640	0.34	67.0	87.2	71.9	73.8	67.7	59.6
+UDP	HRNet-W48	640 × 640	4.2 (×12.4)	67.2 (+0.2)	87.4	72.1	74.5	67.8	59.3
HigherHRNet [26]	HigherHRNet-W48	640 × 640	0.5	65.9	86.4	70.6	73.3	66.5	57.9
+UDP	HigherHRNet-W48	640 × 640	1.8 (×3.6)	66.7 (+0.8)	86.6	71.7	74.2	67.3	59.1
Bottom-up methods with multi-scale ((×2, ×1, ×0.5)) test as in HigherHRNet [26]									
HigherHRNet [26]†	HRNet-W32	512 × 512	-	67.4	87.1	72.3	76.1	67.9	58.6
+UDP	HRNet-W32	512 × 512	-	67.8 (+0.4)	88.0	72.7	76.4	68.3	59.3
HigherHRNet [26]†	HigherHRNet-W32	512 × 512	-	61.4	80.1	65.7	69.9	62.7	50.1
+UDP	HigherHRNet-W32	512 × 512	-	67.5 (+6.1)	87.5	72.5	76.1	68.0	58.8
HigherHRNet [26]†	HRNet-W48	640 × 640	-	68.8	88.3	73.9	76.5	69.5	60.2
+UDP	HRNet-W48	640 × 640	-	69.0 (+0.2)	88.5	74.0	76.9	69.5	60.7
HigherHRNet [26]	HigherHRNet-W48	640 × 640	-	67.6	87.4	72.6	75.8	68.1	58.9
+UDP	HigherHRNet-W48	640 × 640	-	68.2 (+0.6)	88.0	72.9	76.6	68.7	59.9

TABLE 4: Ablation study in top-down paradigm on COCO *val* set. UCST denotes Unbiased Coordinate System Transformation. SNOOP denotes Shift the Network Output by One Pixel. EC denotes Extra Compensation. UKFTCCRF denotes Unbiased Keypoint Format Transformation in Combined Classification and Regression Form, UKFTCF denotes Unbiased Keypoint Format Transformation in Classification Form.

ID	FT	UCST	SNOOP	EC	UKFTCCRF	UKFTCF	AP
A							74.5
B		✓					74.4
C	✓						73.3
D	✓	✓					75.7
E	✓		✓				75.6
F	✓		✓	✓			75.8
G	✓				✓		74.5
H	✓	✓			✓		76.8
I	✓	✓				✓	76.8

result, EC offers just 0.2 AP (75.6→75.8) improvement which is in line with the aforementioned inference. We empirically blame the ineffective of EC for the insensitive of the evaluation system, where the human pose are manually annotated with a certain variance. Proving by EC, the existence of residual error $e(x)'_o$ indicates that the widely used unreported compensation (SNOOP) is a suboptimal remedy not only for its low interpretability, but also for its poorer accuracy.

With configuration E and I, we replace the encoding-decoding methods in configuration D with UKFT-CCRF and UKFT-CF, respectively. With UKFT, configuration E (76.8 AP) and I (76.8 AP) have similar improvement (+1.1 AP) upon baseline configuration D (75.7 AP), which indicates that the biased keypoint format transformation has a considerable impact on the performance. Beside, this also tells that the configuration (i.e., HRNet-W32 network structure with 256×192 input size and training settings in [25]) used in this subsection has similar learning ability on the two unbiased format introduced in this paper. With configuration G where UCST is absent and only UKFT-CCRF is applied, the performance degrades by -2.3 AP to 74.5 AP. Both UCST and UKFT are important for accurate prediction and the defects in the data processing pipeline will have accumulative impact on the

result.

TABLE 5: Ablation study of techniques in bottom-up paradigm on COCO *val* set. HNOR denotes Higher Network Output Resolution, UCST denotes Unbiased Coordinate System Transformation, UKFT-CF denotes Unbiased Keypoint Format Transformation in Classification Form and RNO denotes Resize the Network Output.

ID	HNOR	UCST	UKFT-CF	RNO	IPS	AP
A				✓	0.8	64.4
B		✓			4.9	65.9
C		✓	✓		4.9	67.0
D		✓	✓	✓	0.8	66.1
E	✓				2.9	66.9
F	✓			✓	1.1	67.1
G	✓		✓	✓	1.1	67.1
H	✓	✓			2.9	67.3
I	✓	✓	✓		2.9	67.8
J	✓	✓	✓	✓	1.1	67.8

4.4 Ablation Study on Bottom-up Paradigm

In this subsection, we study how Higher Network Output Resolution (HNOR), Unbiased Coordinate System Transformation (UCST), Unbiased Keypoint Format Transformation in Classification Form (UKFT-CF) and Resize the Network Output (RNO) affect the bottom-up method HigherHRNet [26]. Flipping Testing (FT) is used as default. Experimental settings and the corresponding performance on COCO *val* set are listed in Table 5.

With configuration B, We firstly remove the operation of RNO and apply UCST to the baseline configuration A (64.4 AP and 0.8 IPS). This offers a performance improvement of 1.5 AP and a speed up of 5.9 times to 65.9 AP with 4.9 IPS inference speed. By additionally applying UKFT-CF in configuration B, configuration C scores 67.0 AP with the same inference speed. Both UCST and UKFT are effective as in top-down paradigm.

The referenced configuration F with 67.1 AP and 1.1 IPS inference speed is the recommended settings in HigherHRNet [26]. By constructing configuration E, we remove RNO from it to test the effect of this operation. And according to the result, RNO provides a negligible improvement of 0.2 AP at the high cost of 2.6 times latency in inference. With configuration H and

I, we show that the proposed UCST and UKFT-CF incrementally promote the performance on configuration E by 0.4 AP to 67.3 AP and by additionally 0.5 AP to 67.8 AP, while the inference speed is maintained in 2.9 IPS. These improvements are relatively small when compared with that in configuration B and C. This is in line with the theory that HNOR helps suppress part of the systemic error hidden in data processing pipeline. When unbiased data processing is applying, the HigherHRNet-W32 backbone (i.e., configuration I) still has 0.8 AP superiority on HRNet-W32 (i.e., configuration C) but at the cost of extra 70% latency in inference.

Finally, with configuration D and J, we test the impact of Resize the RNO on the results with UDP. The performance variances are 0 AP with 6.1 times latency and -0.9 AP with 2.6 times latency respectively. RNO is unnecessary for bottom-up paradigm when unbiased data processing is applied. The performance degradation in configuration D from C is derived from the distribution variation caused by the resizing operation. As this destroys the precondition of using UKFT-CF, where a gaussian distribution is strictly required [29].

5 CONCLUSION

In this paper, the common biased data processing for human pose estimation is quantitatively analysed. Interestingly, we find that the systematic errors in standard coordinate system transformation and keypoint format transformation couple together, significantly degrade the performance of human pose estimators in both top-down and bottom-up paradigms. 这为研究界设下了一个陷阱 随后产生了许多无效的补救措施 A trap is laid for the research community and subsequently give birth to many suboptimal remedies. This paper solves this problem by formulating a principled **Unbiased Data Processing (UDP)** strategy, which consists **unbiased coordinate system transformation** and **unbiased keypoint format transformation**. UDP not only pushes the performance boundary of human pose estimation, but also provides a reliable baseline for research community by wiping out the trap formulated in the defective data processing pipeline. 有缺陷的数据处理管道

REFERENCES

- [1] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, jul 2014.
- [2] H. Ci, X. Ma, C. Wang, and Y. Wang, "Locally connected network for monocular 3d human pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [3] D. Luvizon, D. Picard, and H. Tabia, "Multi-task deep learning for real-time 3d human pose estimation and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [4] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3d human pose machines with self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1069–1082, 2020.
- [5] C. Wang, Y. Wang, Z. Lin, and A. L. Yuille, "Robust 3d human pose estimation from single images or video sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1227–1241, 2019.
- [6] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Monocap: Monocular human motion capture using a cnn coupled with a geometric prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 901–914, 2019.
- [7] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person: Joint body parsing & pose estimation network and a new benchmark," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 871–885, 2018.
- [8] S. Park, B. X. Nie, and S. C. Zhu, "Attribute and-or grammar for joint parsing of human pose, parts and attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1555–1569, 2018.
- [9] M. Lu, K. Poston, A. Pfefferbaum, E. V. Sullivan, L. Fei-Fei, K. M. Pohl, J. C. Niebles, and E. Adeli, "Vision-based estimation of mds-updrs gait scores for assessing parkinson's disease motor severity," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds. Cham: Springer International Publishing, 2020, pp. 637–647.
- [10] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using openpose," *Symmetry*, vol. 12, no. 5, p. 744, 2020.
- [11] K. Chen, P. Gabriel, A. Alasfour, C. Gong, W. K. Doyle, O. Devinsky, D. Friedman, P. Dugan, L. Melloni, T. Thesen *et al.*, "Patient-specific pose estimation in clinical environments," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–11, 2018.
- [12] P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang, and G. Huang, "State-aware re-identification feature for multi-target multi-camera tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [13] R. Zhang, Z. Zhu, P. Li, R. Wu, C. Guo, G. Huang, and H. Xia, "Exploiting offset-guided network for pose estimation and tracking," in *CVPR Workshops*, 2019.
- [14] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5167–5176.
- [15] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.
- [16] J. Carreira and A. Zisserman, "Quo vadis, action recognition a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.
- [17] J. Zhu, W. Zou, Z. Zhu, L. Xu, and G. Huang, "Action machine: Toward person-centric action recognition in videos," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1633–1637, 2019.
- [18] J. Zhu, W. Zou, Z. Zhu, and Y. Hu, "Convolutional relation network for skeleton-based action recognition," *Neurocomputing*, vol. 370, pp. 109–117, 2019.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
- [21] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.
- [22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [23] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.
- [24] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *European Conference on Computer Vision*, 2018, pp. 466–481.
- [25] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*.
- [26] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5386–5395.
- [27] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7103–7112.
- [28] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *arXiv preprint arXiv:1901.00148*, 2019.
- [29] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proceedings*

- of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7093–7102.
- [30] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, “Crowdpose: Efficient crowded scenes pose estimation and a new benchmark,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10863–10872.
 - [31] J. Huang, Z. Zhu, F. Guo, and G. Huang, “The devil is in the details: Delving into unbiased data processing for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5700–5709.
 - [32] J. Huang, Z. Shan, Y. Cai, F. Guo, Y. Ye, X. Chen, Z. Zhu, G. Huang, J. Lu, and D. Du, “Joint coco and lvis workshop at eccv 2020: Coco keypoint challenge track technical report: Udp+,” in *ECCV Workshop*, 2020.
 - [33] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4733–4742.
 - [34] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1014–1021.
 - [35] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.
 - [36] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
 - [37] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.
 - [38] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1799–1807.
 - [39] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.
 - [40] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.
 - [41] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, “Learning feature pyramids for human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
 - [42] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, “Whole-body human pose estimation in the wild,” in *European Conference on Computer Vision*. Springer, 2020, pp. 196–214.
 - [43] U. Iqbal and J. Gall, “Multi-person pose estimation with local joint-to-person associations,” in *European Conference on Computer Vision*. Springer, 2016, pp. 627–642.
 - [44] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, “Deepcut: Joint subset partition and labeling for multi person pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4929–4937.
 - [45] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, “Deepcut: A deeper, stronger, and faster multi-person pose estimation model,” in *European Conference on Computer Vision*, 2016, pp. 34–50.
 - [46] Z. Cao, G. Hidalgo Martínez, T. Simon, S. Wei, and Y. A. Sheikh, “Openpose: Realtime multi-person 2d pose estimation using part affinity fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
 - [47] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, “Towards accurate multi-person pose estimation in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4903–4911.
 - [48] X. Nie, J. Feng, J. Zhang, and S. Yan, “Single-stage multi-person pose machines,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6951–6960.
 - [49] A. Newell, Z. Huang, and J. Deng, “Associative embedding: End-to-end learning for joint detection and grouping,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2277–2287.
 - [50] T. Golda, T. Kalb, A. Schumann, and J. Beyerer, “Human pose estimation for real-world crowded scenarios,” in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, pp. 1–8.
 - [51] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, “Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model,” in *European Conference on Computer Vision*, 2018, pp. 269–286.
 - [52] Z. Su, M. Ye, G. Zhang, L. Dai, and J. Sheng, “Cascade feature aggregation for human pose estimation,” *arXiv preprint arXiv:1902.07837*, 2019.
 - [53] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.
 - [54] M. Kocabas, S. Karagoz, and E. Akbas, “Multiposenet: Fast multi-person pose estimation using pose residual network,” in *European Conference on Computer Vision*, 2018, pp. 417–433.
 - [55] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
 - [56] Y. Chen, C. Shen, H. Chen, X. Wei, L. Liu, and J. Yang, “Adversarial learning of structure-aware fully convolutional networks for landmark localization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1654–1669, 2020.
 - [57] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.
 - [58] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
 - [59] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang, “Multi-person pose estimation with enhanced channel-wise and spatial information,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5674–5682.
 - [60] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, “Integral human pose regression,” in *European Conference on Computer Vision*, 2018, pp. 529–545.
 - [61] S. Kreiss, L. Bertoni, and A. Alahi, “Pifpaf: Composite fields for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11977–11986.
 - [62] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.
 - [63] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, “Integral human pose regression,” in *European Conference on Computer Vision*, 2018, pp. 529–545.
 - [64] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
 - [65] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “Rmpe: Regional multi-person pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.
 - [66] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, “Learning feature pyramids for human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1281–1290.
 - [67] S. Huang, M. Gong, and D. Tao, “A coarse-fine network for keypoint localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3028–3037.
 - [68] G. Moon, J. Y. Chang, and K. M. Lee, “Posefix: Model-agnostic general human pose refinement network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7773–7781.
 - [69] D. Yu, K. Su, X. Geng, and C. Wang, “A context-and-spatial aware network for multi-person pose estimation,” *arXiv preprint arXiv:1905.05355*, 2019.
 - [70] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang et al., “Hybrid task cascade for instance segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.