Lightweight Super-Resolution Head for Human Pose Estimation

Haonan Wang
State Key Laboratory for Novel Software Technology,
Nanjing University,
Nanjing, China
wanghaonan0522@gmail.com

Jie Tang
State Key Laboratory for Novel Software Technology,
Nanjing University,
Nanjing, China
tangjie@nju.edu.cn

Jie Liu*
State Key Laboratory for Novel Software Technology,
Nanjing University,
Nanjing, China
liujie@nju.edu.cn

Gangshan Wu
State Key Laboratory for Novel Software Technology,
Nanjing University,
Nanjing, China
gswu@nju.edu.cn

ABSTRACT

Heatmap-based methods have become the mainstream method for pose estimation due to their superior performance. However, heatmap-based approaches suffer from significant quantization errors with downscale heatmaps, which result in limited performance and the detrimental effects of intermediate supervision. Previous heatmap-based methods relied heavily on additional post-processing to mitigate quantization errors. Some heatmap-based approaches improve the resolution of feature maps by using multiple costly upsampling layers to improve localization precision. To solve the above issues, we creatively view the backbone network as a degradation process and thus reformulate the heatmap prediction as a Super-Resolution (SR) task. We first propose the SR head, which predicts heatmaps with a spatial resolution higher than the input feature maps (or even consistent with the input image) by super-resolution, to effectively reduce the quantization error and the dependence on further post-processing. Besides, we propose SRPose to gradually recover the HR heatmaps from LR heatmaps and degraded features in a coarse-to-fine manner. To reduce the training difficulty of HR heatmaps, SRPose applies SR heads to supervise the intermediate features in each stage. In addition, the SR head is a lightweight and generic head that applies to top-down and bottom-up methods. Extensive experiments on the COCO, MPII, and Crowd-Pose datasets show that SRPose outperforms the corresponding heatmap-based approaches. The code and models are available at https://github.com/haonanwang0522/SRPose.

*Corresponding author

1 INTRODUCTION

2D human pose estimation (HPE) is one of the fundamental tasks in computer vision [6]. Its purpose is to localize all the human anatomy keypoints from a single image. Since it is the basis for many human-centric visual understanding tasks such as 3D human pose estimation [49, 53, 11, 43], human action recognition [41, 15, 2] and pose tracking [12, 42, 45], it has attracted widespread attention from academia and industry.

2D HPE can be broadly classified into two frameworks: top-down [35, 45] and bottom-up [28, 9]. Top-down methods first detect human instances by detectors and then locate the keypoints of each human instance. However, bottom-up methods pinpoint human keypoints at first and then assign them to different human instances. In recent years, heatmap-based methods have achieved superior performance in both, especially in the top-down framework, which has become the *de facto* standard. Specifically, the heatmap-based methods generate a heatmap for each kind of keypoints. Each heatmap contains a 2D Gaussian distribution centered at the ground-truth joint position, suppressing false positives and smoothing the training process. As a result, the heatmap-based methods have a more robust generalization and are easier to optimize than those that directly regress coordinates.
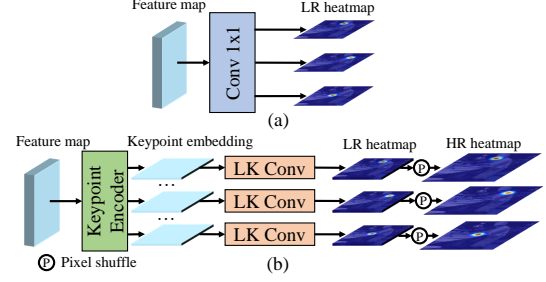
Though achieving superior performance, the heatmap-based methods suffer from non-negligible quantization errors since continuous coordinate values need to be discretized into downscale heatmaps, and the resolution of the heatmaps limits the coordinate precision. The presence of quantization errors makes numerous heatmap-based methods achieve poor performance when decoding directly. Most

Figure 1: Comparison of SRPose with previous heatmap-based methods. (a) shows the method that upsamples low-resolution feature maps by heavily upsampling layers. (b) shows the method that learns high-resolution feature maps directly. (c) shows the method that learns horizontal 1D heatmaps and vertical 1D heatmaps. (d) shows our method to obtain heatmaps with $H \times W$ resolution from heatmaps with $\frac{H}{32} \times \frac{W}{32}$ resolution by SR Head.

Figure 2: Comparison of SR head and Simple head. (a) illustrates the simple head. (b) illustrates the SR head, where 'LK Conv' indicates large kernel convolution.

heatmap-based methods rely heavily on further post-processing (*e.g.*, empirical shifts, DARK [50]) to mitigate quantization errors, but such post-processing is poorly optimized. In addition, some methods alleviate quantization errors by increasing the resolution of the heatmaps, and we summarize three typical ways in Fig. 1. As shown in Fig. 1(a), [45, 46] extract Low-Resolution (LR) features (*e.g.*, $\frac{H}{32} \times \frac{W}{32}$) with rich semantic information via the backbone network. The resolution is then increased to $\frac{H}{4} \times \frac{W}{4}$ by upsampling layers. Finally, the High-Resolution (HR) features are fed into a simple prediction head (usually a 1×1 convolution) to obtain heatmaps with $\frac{H}{4} \times \frac{W}{4}$ resolution.

However, the computational overhead of the upsampling layers is usually huge (*e.g.*, costly deconvolution layer in [45, 46]) to achieve better performance. As shown in Fig. 1(b), [35, 48] directly extract HR features from the input images and feed them into a simple prediction head to yield keypoint heatmaps. However, to prevent excessive computational overhead, the finest resolution of the intermediate feature is only $\frac{H}{4} \times \frac{W}{4}$. When mapping back to the original image, there is still a large quantization error. This paper aims to mitigate quantization errors with higher resolution without excessive computational overheads. Recently, the SimCC [20] approach uniformly divides each pixel into several bins and reformulates HPE as two classification tasks for horizontal and vertical coordinates, as shown in Fig. 1(c). In this way, SimCC achieves sub-pixel localization precision and low quantization error. Nevertheless, SimCC is

hard to be applied to bottom-up methods directly because there may be multiple instances for each kind of joint.

In this paper, we creatively view the backbone network as a degradation process and thus reformulate the heatmap prediction as a Super-Resolution (SR) task. So we can learn from the successful experience in the SR task to design a lightweight SR head for HR heatmap generation. Specifically, as shown in Fig. 2(b), the SR head first encodes all kinds of keypoints to keypoint embeddings by the keypoint encoder. For the ith keypoint embedding K_i, it is decoded by separate large kernel convolution to compute multiple heatmaps for the ith keypoint, and finally stitch all the ith keypoint's heatmaps together by pixel shuffle [34] to obtain the HR heatmap for the ith keypoint. Further, although the backbone network is a degradation process, the semantic information becomes richer. Therefore, we propose SRPose to predict LR heatmaps first from LR feature maps and then use the LR heatmaps and degraded features to recover HR heatmaps from coarse to fine (see Fig. 1(d)). Besides, during training, the application of SR heads can reduce the difficulty of obtaining high-resolution heatmaps. During inference, we only keep the last SR head to improve efficiency.

Compared with previous works, our design has three merits. (1) SR head is simple, lightweight, and suitable for bottom-up and top-down frameworks. (2) SR head can be employed to predict high-resolution heatmaps (the resolution is consistent with the input), in this way to reduce quantization errors effectively. (3) SR head can be deployed for high-resolution supervision, thus alleviating the problem that low-resolution features are difficult to supervise.

The contributions are summarized as follows:

- We propose a lightweight head, called SR head, that can predict high-resolution heatmaps to improve the spatial resolution of heatmaps to reduce quantization errors. Besides, As a generic regression head, the SR head can be applied to top-down and bottom-up methods.
- We propose SRPose, which treats the backbone network as a degradation process to reformulate the pose estimation task as a super-resolution task. To reduce the learning difficulty of the HR heatmap, we recover the precise HR heatmaps from the degraded features and LR heatmaps in a coarse-to-fine manner and apply SR heads for supervision.
- We conduct comprehensive experiments on COCO, MPII, and CrowdPose datasets to verify the effectiveness of the SR head and SRPose with different backbones.

2 RELATED WORK

Top-down method. The top-down method involves detecting bounding boxes by object detectors [33, 8] and estimating keypoints for each human instance. The top-down approach can be classified into regression-based [39, 5, 36, 30, 27] and heatmap-based methods [29, 7, 45, 14]. Heatmap-based methods estimate the likelihood of each keypoint for each pixel, which has better performance and is currently dominant. [3] and [19] perform multi-stage feature extraction on the input and supervise the intermediate features. The intermediate supervision is performed on the interpolated low-resolution heatmaps. [45] proposes a simple baseline with three deconvolutions after the backbone network to predict the heatmap. [35] proposes to maintain high-resolution representations throughout the process. [48] introduces Transformer [40] on HRNet to incorporate the self-attentive mechanism. However, quantization error has been a significant problem for heatmap-based methods.

Bottom-up method. The bottom-up methods [28, 9] first detect all unidentified body joints in the input and then group them. [4] learns a 2D vector field, called partial affinity field, connecting two keypoints and groups keypoints based on the integration of the line between the two keypoints. [31] learns the 2D offset field of each pair of keypoints to group the keypoints. [28] proposes associative embedding for grouping, which assigns a tag to each keypoint and groups the keypoints according to the l2 distance between the tag vectors. Following it, [9] learns a higher resolution feature pyramid based on HRNet to improve the precision of small people.

Quantization error. Heatmap-based methods have seriously problematic quantization errors due to discretization and low resolution. To alleviate it, [50] uses a post-processing approach to synthesize the distribution information of heatmap activation by Taylor expansion-based distribution approximation. [20] treats the pose estimation task as a classification task and tries to predict two decoupled high-resolution 1D heatmaps to improve the localization precision. However, the head of SimCC is MLP which still requires heavy weights. Besides, it applies exclusively to top-down methods, while it is difficult to apply to bottom-up methods because a heatmap contains more than one keypoint.

Super-resolution in pose estimation. There have been previous works [51, 25] combining pose estimation tasks with super-resolution tasks. However, they recover the LR input through a super-resolution network to obtain an HR human image. The HR human image is later fed into the pose estimation network to estimate heatmaps. As a result, most of their inputs are of low resolution, as a higher resolution input would entail expensive computational overhead. Therefore, it is not suitable for mitigating quantization errors with this approach. Our method first proposes the SR head, which enables the model to learn HR heatmaps from LR feature maps. Besides, we treat the pose estimation task as a super-resolution task. Therefore, we recover the HR heatmaps from LR heatmaps and features in a coarse-to-fine manner. Meanwhile, we perform stage-by-stage HR supervision through SR heads for more precise recovery.

3 METHOD

The key idea of our method is to generate HR heatmaps from LR heatmaps and features to alleviate quantization errors. As shown in Fig. 3(a), we propose a general training and inference framework

dubbed SRPose for more accurate HPE. SRPose mainly consists of a backbone for feature extraction and LR heatmap generation, a neck for hierarchical feature fusion, and multiple SR heads for HR heatmap prediction and supervision.

3.1 Architecture of SRPose

Backbone. SRPose can be applied to both Transformer-based and CNN-based backbones. In this paper, we creatively view the backbone as an image degradation model. For an input image $I \in \mathbb{R}^{H \times W \times 3}$, the backbone can extract features and generate LR heatmaps as follows:

$$F_2, F_3, F_4, F_5, M_{LR} = \mathcal{D}(I), \qquad (1)$$

where $\mathcal{D}$ denotes the degradation process through the backbone, F_k represents the feature which has strides of 2^k pixels with respect to the input image and M_{LR} is the LR heatmaps. After obtaining the LR heatmaps, we design a novel SRPose to generate HR heatmaps effectively. The core part of the proposed SRPose is the SR head which will be described in detail next.

SR Head. LR heatmaps lead to significant quantization errors. During decoding, the precision of HPE heavily depends on the resolution of heatmaps. In existing HPE networks, the resolution of heatmaps has already reached $\frac{H}{4} \times \frac{W}{4}$, while the quantization error is still unacceptable. Except for the final heatmaps, the supervision of intermediate features also suffers from quantization errors. For supervision, [3, 19] generate LR heatmaps from LR intermediate features. The resolution of heatmaps is extended to $\frac{H}{4} \times \frac{W}{4}$ by corner-aligned interpolation. Although the resolution is increased by interpolation, the precision is still limited by the original heatmap. To alleviate the above problems, we propose to generate multiple heatmaps for each keypoint, and then use a pixel-shuffle [34] layer to combine multiple LR heatmaps into an HR heatmap. Specifically, as shown in Fig. 3(c), we extract several embeddings for each keypoint from the feature by the keypoint encoder as:

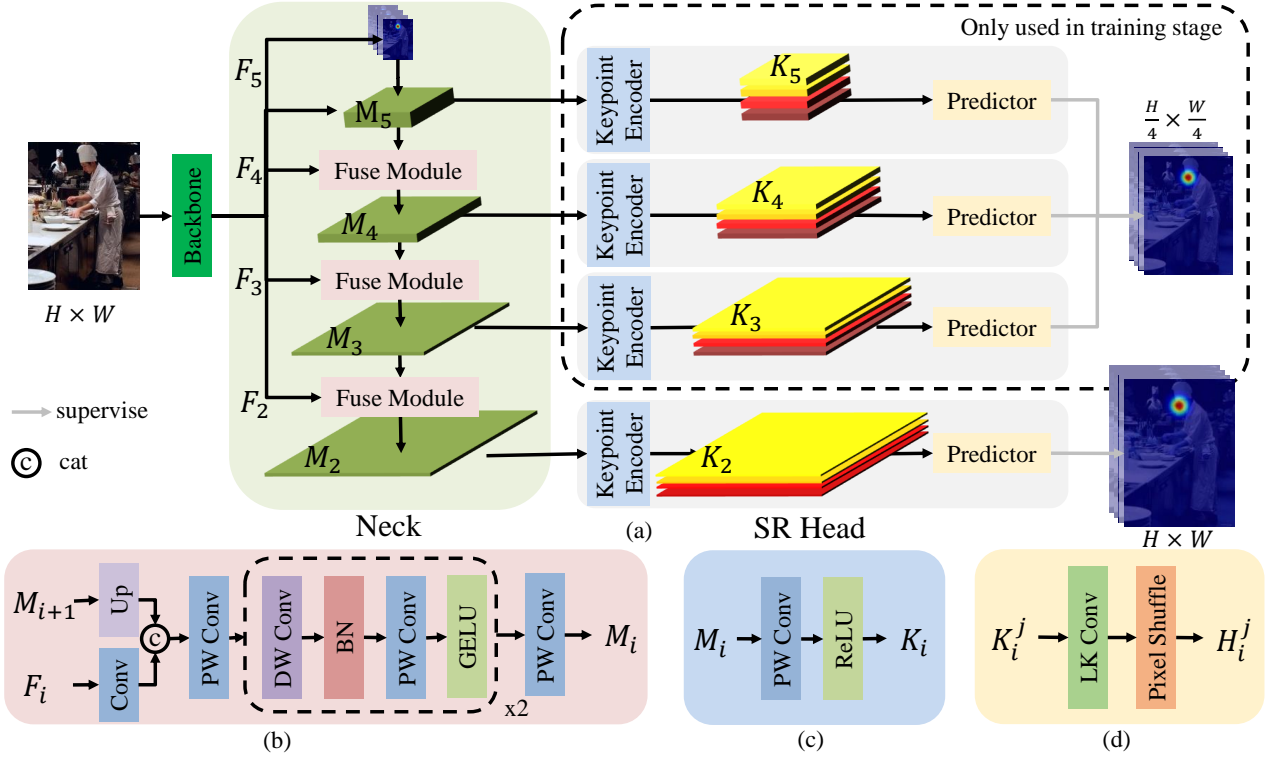$$K^0 \cdots K^{N-1} = \text{Encoder}(M), \qquad (2)$$

where M is the feature to be decoded, and K denotes the embedding of the keypoint. There are N keypoints in total. Then, each keypoint feeds its embedding into the corresponding Large Kernel Convolution (LKC) layer to generate heatmaps as:

$$h^j = \text{LKC}^j(K^j), \qquad (3)$$

where h^j donates the LR heatmaps for the j-th keypoint. h^j contains C heatmaps and the number C is determined by the upsampling factor of the heatmaps. Specifically, if an HR heatmap with $(l \cdot h) \times (l \cdot w)$ resolution is obtained from a feature map with $h \times w$ resolution, $C = l^2$ LR heatmaps need to be generated for each keypoint. LKC^j denotes the convolution layer for the j-th keypoint. We set large kernels in LKC to better discriminate keypoints to capture larger perceptual fields. And then, we combine LR heatmaps to form an HR heatmap via pixel shuffle as follows:

$$H^j = \text{PixelShuffle}(h^j, l), \qquad (4)$$

where H^j denotes the HR heatmap for the j-th keypoint and l denotes the upsampling factor of pixel shuffle. The details of the Pixel Shuffle layer are shown in the Fig. 4. In this way, we can convert the information in the channels to spatial resolution. As a result, we can obtain HR heatmaps without HR feature maps. Therefore, we

Figure 3: (a) Overview of SRPose architecture. (b) The fuse module. (c) The keypoint encoder. (d) The predictor. In SRPose, the backbone is viewed as an image degradation model, and we first generate an LR heatmap by a CNN-based or Transformer-based backbone. We employed SR heads in a coarse-to-fine manner to generate HR heatmaps effectively during the training process. During inference, only the last SR head is kept for highly efficient.

Figure 4: Schematic diagram of the Pixel Shuffle layer.

do not have to extract HR features by deconvolution to reduce the parameters and computational overhead effectively. In addition, each HR heatmap is generated from its keypoint embedding, which makes the SR head more lightweight even with large kernel convolution due to the small number of channels in the keypoint embedding.

HR Supervision. The proposed SR head needs to increase the LR heatmaps (*e.g.*, $\frac{H}{32} \times \frac{W}{32}$) to the final HR heatmaps (*e.g.*, $H \times W$). However, optimizing the network at such a significant scale is extremely difficult. To alleviate the training difficulty, we propose to utilize multiple SR heads in the SRPose. As shown in Fig. 3(a), the SR-Pose contains four SR heads, among which the first three heads are responsible for HR supervision, and the last head is designated for final HR heatmaps. In other words, the first three SR heads are only used in the training stage and will be removed during inference. As we have discussed, current supervision methods actually supervise

the intermediate feature at a relatively low resolution. Different from these methods, our SR head achieves HR supervision at a spatial resolution of $\frac{H}{4} \times \frac{W}{4}$.

As shown in Fig. 3(a), we adopt an FPN-like [22] neck to generate input features for each SR head. Specifically, we feed the extracted feature maps from the backbone into feature fusion modules to form more representative features for SR heads:

$$M_i = \text{Fusion}(F_i, M_{i+1}), \tag{5}$$

where i iterates from 5 to 2, and M_6 is the LR heatmaps (*i.e.*, M_{LR} in Eq. (1)). M_i serves as the input for the corresponding SR head, which is formally described in Eq. (2). When $i = 5$, Fusion(.) denotes convolution, and when $i \neq 5$, Fusion(.) denotes the "Fuse Module" which is shown in Fig. 3(b).

Bottom up. The bottom-up approaches not only locate keypoints but also group keypoints to each instance, so we only replace the simple head in the baseline with the SR head to generate HR heatmaps. We use the simple method of associative embedding to perform the grouping. Associate embeddings are generated from the feature with $\frac{H}{4} \times \frac{W}{4}$ resolution by 1×1 convolution. To unify the resolution, we interpolate the associate embeddings:

$$ae = \text{Conv}_{1 \times 1}(F_2),$$
$$AE = \text{Interpolate}(ae, l), \tag{6}$$

Figure 5: Overview of the our inference process, which reduces inference time without compromising precision via max pooling.

where ae donates the LR associate embeddings, AE donates the HR associate embeddings, and l denotes the scale factor between the LR associate embeddings and the HR heatmaps.

3.2 Training Targets and Loss Functions

Heatmaps indicate the coordinates of keypoints with spatial confidence. It is generally designed to follow an unnormalized Gaussian distribution:

$$G(\boldsymbol{c}; \boldsymbol{\mu}, \sigma) = \exp(-\frac{(\boldsymbol{c} - \boldsymbol{\mu})^{\mathrm{T}}(\boldsymbol{c} - \boldsymbol{\mu})}{2\sigma^2}), \tag{7}$$

where $\boldsymbol{c}$ denotes the coordinates of the heatmap pixel (c_x, c_y), $\boldsymbol{\mu}$ denotes the coordinates of the target joint (μ_x, μ_y), σ denotes the pre-defined constant. To make the before-and-after supervision consistent, we set σ as linear to the resolution. If $\sigma = 2$ for heatmaps with resolution $\frac{H}{4} \times \frac{W}{4}$, then $\sigma = 8$ for heatmaps with resolution $H \times W$. For all heatmaps, we regard the mean square error (MSE) as the loss function for supervision.

3.3 Inference

We generate heatmaps with higher resolution, which effectively alleviates the problem of quantization errors, but also increases inference time. Because of the finer granularity of the candidate regions, we have to find the global maximum from more candidate regions. To alleviate the problem of our long inference time, we optimize the inference process of the heatmap. Coordinates decoding in previous heatmap-based methods is computed serially on the CPU, leading to considerable inference latency. We design a new decoding paradigm to reduce the long decoding time without influencing inference accuracy.

Briefly, we adopt the idea of divide-and-conquer, first dividing the image into multiple non-overlap patches, and then parallelizing the computation in each patch by GPU to get the maximum value of each patch and the location of the maximum value, and then finding the maximum value among all the great values. The original location of the maximum value is the final location. Specifically, as shown in Fig. 5, we utilize max pooling to reduce the resolution of heatmaps.

$$\boldsymbol{Max}_v, \boldsymbol{Max}_i = \text{MaxPooling}_s(H_2), \tag{8}$$

where $\boldsymbol{Max}_v$ and $\boldsymbol{Max}_i$ denote the maximum value and corresponding index of each patch. MaxPooling$_s$ denotes the max-pooling layer whose kernel size and stride are both s. To reduce the inference time, instead of decoding on the original heatmap, we decode $\boldsymbol{Max}_v$ to get the coarse location:

$$\boldsymbol{Loc}_{coarse} = \text{Decoder}(\boldsymbol{Max}_v). \tag{9}$$

Finally, the fine location is gathered by the index of $\boldsymbol{Max}_i$ according to the coarse location:

$$\boldsymbol{Loc}_{fine} = \boldsymbol{Max}_v[\boldsymbol{Loc}_{coarse}]. \tag{10}$$

Note that the decoding process can be parallelized on GPU devices, and the decoding speed is increased without compromising accuracy. Because we reduce the spatial resolution by max pooling, and we also keep the position of the maximum value in each patch in the original HR heatmap. Finally, we can get the global maximum value and its position in the original HR heatmap, and it does not affect the precision of inference. The detailed results are shown in Fig. 7 of Section 4.2.

4 EXPERIMENTS

In the following sections, we conduct experiments on several datasets to validate the effectiveness of SRPose and SR head for human pose estimation. Experiments are conducted on three benchmark datasets: COCO [23], MPII [1], and CrowdPose [17]. The ablation experiments are conducted on the benchmark dataset of COCO.

4.1 COCO Keypoint Detection

Dataset. The COCO dataset [23] contains more than $200,000$ images and $250,000$ human instances which are labeled with 17 keypoints. It is divided into three sets, the train set is with $57k$ images, the val set is with $5k$ images, and the test-dev set is with $20k$ images. Experimental results are reported on both the test-dev set and the val set. The data augmentation settings follow MMPose [10].

Evaluation metric. The standard evaluation metric for the COCO dataset is the standard average precision(AP), which is based on Object Keypoint Similarity(OKS):

$$OKS = \frac{\sum_i \exp(-d_{p^i}^2/2S_p^2\sigma_i^2)\delta(v_{p^i} > 0)}{\sum_i \delta(v_{p^i} > 0)}, \tag{11}$$

where d_{p^i} indicates the Euclidean distance between i^{th} keypoint of human p and the corresponding ground truth, v_{p^i} indicates visibility of the keypoint, S_p indicates the scale factor of human p, and σ_i indicates the factor of keypoint i.

Backbone settings. Currently, there are many backbones for human pose estimation. For top-down methods, the backbone can be broadly classified into CNN-based and Transformer-based. To demonstrate the applicability of SRPose to both types of backbone, we choose two state-of-the-art methods (*i.e.*, Resnet [13] and HRNet [35]) from the CNN-based methods and two (i.e. TransPose [47] and HRFormer [48]) from the Transformer-based methods as baselines. For bottom-up methods, we take Resnet and HRNet as the backbone.

Implementation details. For the selected backbones, we simply follow the settings in MMPose. Specifically, for top-down methods, all the models are trained with batch size 128 (batch size 128 for HRFormer-B due to limited GPU memory) and are optimized by Adam (AdamW [24] for HRFormer in MMPose) with a base learning rate of 5×10^{-4} decreased to 5×10^{-5} at the 170th epoch, to 5×10^{-6} at the 200th epoch and ended at the 210th; β_1 and β_2 are set to 0.9 and 0.999; weight decay is set to 10^{-4}. Meanwhile, they use the two-stage top-down human pose estimation as the pipeline, which first detects the human by a detector and then crops it out to estimate its keypoints. Following [45], we adopt the person detector with

Table 1: Comparison with different heatmap-based methods on the COCO val set. We conduct experiments on both top-down and bottom-up methods. We use the same human detection frame for a fair comparison of the top-down methods. We tested the results with (w/ Post) and without (w/o Post) adding an empirical shift for each top-down method. The size of the input for all top-down methods is 256×192, while the size of the input for bottom-up methods is 512×512. Our method provides significant gains for all backbones and reduces the dependence on refinement post-processing.

Backbone	Scheme	GFLOPs	Params		w/ Post.		w/o Post.	
			Backbone	Other	AP	AR	AP	AR
Top-down methods								
Resnet-50 [13]	Simple head	5.46	23.51M(69%)	10.49M(31%)	71.7(↓ 0.0)	77.3(↓ 0.0)	69.8(↓ 1.9)	75.8(↓ 1.5)
	SR head	5.77	23.51M(69%)	10.59M(31%)	72.4(↑ 0.7)	77.9(↑ 0.6)	72.2(↑ 0.5)	77.7(↑ 0.4)
	SRPose	4.61	23.51M(95%)	1.29M(5%)	**73.3**(↑ 1.6)	**78.8**(↑ 1.5)	73.1(↑ 1.4)	78.6(↑ 1.3)
HRNet-W32 [35]	Simple head	7.70	28.54M(100%)	0.00M(0%)	74.5(↓ 0.0)	79.9(↓ 0.0)	72.3(↓ 2.2)	78.2(↓ 1.7)
	SR head	7.98	28.54M(100%)	0.09M(0%)	75.6(↑ 1.1)	80.6(↑ 0.7)	75.4(↑ 0.9)	80.5(↑ 0.6)
	SRPose	8.28	29.30M(98%)	0.65M(2%)	**75.9**(↑ 1.4)	**81.0**(↑ 1.1)	75.7(↑ 1.2)	80.9(↑ 1.0)
TransPose-R-A4 [47]	Simple head	8.91	4.93M(82%)	1.06M(18%)	71.8(↓ 0.0)	77.3(↓ 0.0)	69.7(↓ 2.1)	75.5(↓ 1.8)
	SR head	9.23	4.93M(81%)	1.16M(19%)	73.2(↑ 1.4)	78.4(↑ 1.1)	73.1(↑ 1.3)	78.3(↑ 1.0)
	SRPose	6.26	4.93M(90%)	0.55M(10%)	**73.5**(↑ 1.7)	**78.9**(↑ 1.6)	73.4(↑ 1.6)	78.7(↑ 1.4)
HRFormer-S [48]	Simple head	2.82	7.89M(100%)	0.00M(0%)	74.0(↓ 0.0)	79.2(↓ 0.0)	72.1(↓ 1.9)	77.6(↓ 1.6)
	SR head	3.09	7.89M(99%)	0.09M(1%)	75.0(↑ 1.0)	80.1(↑ 0.9)	74.8(↑ 0.8)	80.0(↑ 0.8)
	SRPose	3.34	8.21M(93%)	0.65M(7%)	**75.6**(↑ 1.6)	**80.7**(↑ 1.5)	75.5(↑ 1.5)	80.6(↑ 1.4)
Bottom-up methods								
Resnet-50 [13]	Simple head	29.20	23.51M(69%)	10.49M(31%)	46.7(↓ 0.0)	55.1(↓ 0.0)	-	-
	SR head	30.86	23.51M(69%)	10.60M(31%)	**48.4**(↑ 1.7)	**56.6**(↑ 1.5)	-	-
HRNet-W32 [35]	Simple head	41.10	28.54M(100%)	0.00M(0%)	65.3(↓ 0.0)	70.9(↓ 0.0)	-	-
	SR head	42.57	28.54M(100%)	0.09M(0%)	**67.1**(↑ 1.8)	**71.7**(↑ 0.8)	-	-

56.4% for COCO val set. For bottom-up methods, all the models are trained with batch size 48 and are optimized by Adam with a base learning rate of 1.5×10^{-3} dropped to 1.5×10^{-4} at the 200th epoch, to 1.5×10^{-5} at the 260th epoch and ended at the 300th.

Results on the COCO val set. We conduct extensive experiments based on different backbones on the COCO val set to validate the effectiveness of our proposed SR head and SRPose framework. For top-down methods, some well-performing CNN-based and Transformer-based methods are selected as the baseline, while the bottom-up methods only chose CNN-based methods as the baseline due to the oversized inputs. The results in Table 1 illustrate that the SR head is efficient and shows consistent performance dominance over the Simple head across different backbones and frameworks, especially for the bottom-up methods. For instance, in the bottom-up HRNet-W32 [9], replacing Simple head with SR head results in a gain of 1.9 AP, but the number of parameters is increased by only 0.09M. In addition, the SR head reduces the dependence of the top-down method on post-processing due to the HR heatmap. For example, when HRNet-W32 [35] serves as the backbone, SR head drops only 0.2AP without post-processing, but Simple head drops 2.2AP. Overall, whether the backbone is pyramid-based or HR-based, SRPose

can bring significant benefits with reduced or small additional parameters. For example, in ResNet-50 [13], SRPose improves 1.6AP, while in HRFormer-S, SRPose improves 1.6AP.

Results on the COCO test-dev set. We perform the comparison on the COCO test-dev set, with the results shown in Table 2. For top-down methods with 384×288 input size, our method boosts 2.6AP and 0.7AP for SimpleBaseline-Res50 [45] and HRFormer-B [48], respectively. For bottom-up methods with 512×512 input size, our method improves 2.4AP for HRNet-W32, even better than HigherHRNet [9] with fewer GFOLPs.

4.2 Ablation Study

In this section, we use Resnet-50 as the backbone to conduct ablation studies for investigating our proposed model. The training settings are the same as Section 4.1, and we provide the test results of COCO val to observe the performance. The default resolution is 256×192.

Scale factor k. Scale factor k represents the resolution ratio of the output heatmap to the input image. It is proportional to the resolution of the output heatmap and inversely proportional to quantization errors. However, it is also proportional to the number of parameters in the SR head, which may lead to overfitting easily. To balance between the model performance and quantization error, we set $k \in$

Table 2: Comparison on COCO test-dev set. 'T' denotes the abbreviation of Transformer. Our method achieves state-of-the-art results with significantly improved performance compared to the baseline.

Method	Backbone	Input size	GFLOPs	AP	AR
Top-down methods					
Regression-based					
CenterNet [52]	Hourglass	-	-	63.0	-
DirectPose [38]	ResNet-50	-	-	62.2	-
PointSetNet [44]	HRNet-W48	-	-	68.7	-
Integral Pose [37]	ResNet-101	256×256	11.0	67.8	-
TFPose [26]	ResNet-50+T	384×288	20.4	72.2	-
RLE [16]	HRNet-W48	-	-	75.7	-
Heatmap-based					
SimBa [45]	ResNet-50	384×288	20.0	71.5	76.9
SimBa [45]	ResNet-152	384×288	35.6	73.7	79.0
TransPose [47]	HRNet-W48+T	256×192	21.8	75.0	80.1
TokenPose [21]	L/D24	384×288	22.1	75.9	80.8
HRNet [35]	HRNet-W32	384×288	16.0	74.9	80.1
HRNet [35]	HRNet-W48	384×288	32.9	75.5	80.5
SimCC [20]	ResNet-50	384×288	20.2	72.7	78.0
SimCC [20]	HRNet-W48	384×288	32.9	76.0	81.1
DARK [50]	HRNet-W48	384×288	32.9	76.2	81.1
HRFormer [48]	HRFormer-S	384×288	6.2	74.5	79.8
HRFormer [48]	HRFormer-B	384×288	26.8	76.2	81.2
SRPose	ResNet-50	384×288	24.6	74.1	79.1
SRPose	HRFormer-B	384×288	30.4	**76.9**	**81.8**
Bottom-up methods					
OpenPose [4]	-	-	-	61.8	66.5
AE [28]	Hourglass	512×512	206.9	65.5	70.2
HRNet [9]	HRNet-W32	512×512	38.9	64.1	-
HigherHRNet [9]	HRNet-W32	512×512	47.9	66.4	-
HRNet + SR head	HRNet-W32	512×512	42.6	**66.5**	**71.2**

Table 3: Ablation study of scale factor k on the COCO val set. k controls the resolution ratio of the output heatmap to the input image, and it improves performance at the beginning of growth but then the performance decreases.
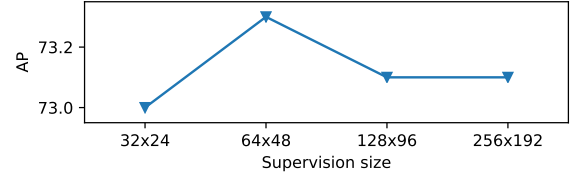
k	Output size	Params	GFLOPs	AP	AR
0.25	64×48	24.71M	4.36	72.7	78.2
0.5	128×96	24.73M	4.41	73.1	78.6
1	256×192	24.80M	4.61	**73.3**	**78.8**
2	512×384	25.06M	5.43	73.2	78.6

$\{0.25, 0.5, 1, 2\}$ to experiment. According to Table 3, it is easy to find that when k grows from 0.25 to 0.5, the model's performance has a large improvement, but as k grows, the performance growth slows down. The model reaches its best when $k = 1$, thus we set $k = 1$ in SRPose.
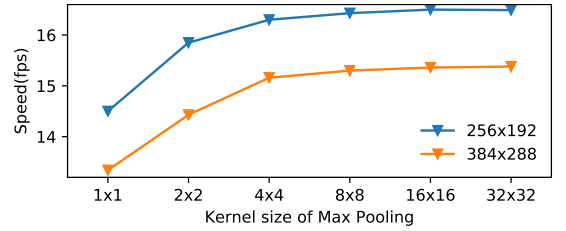
HR supervision. To verify the necessity of HR supervision, we conduct experiments by gradually reducing HR supervision from top to down. For better comparison, we regard SimpleBaseline + SR head as the baseline. As shown in Table 4, Simply fusing the features can only bring a slight improvement to the model performance, but as the number of supervisions grows, so does the model's performance. And from the results, we can find that it is necessary to supervise the topmost feature, which can significantly improve performance.

Table 4: Ablation study of HR supervision on the COCO val set. As supervision decreases, so does performance.

Method	Supervised Feature				AP	AR
	M_5	M_4	M_3	M_2		
SimBa + SR head	✗	✗	✗	✓	72.4	77.9
SRPose	✓	✓	✓	✓	**73.3**	**78.8**
	✗	✓	✓	✓	72.7	78.2
	✗	✗	✓	✓	72.7	78.2
	✗	✗	✗	✓	72.6	78.1

Figure 6: Ablation study of supervision size on the COCO val set. The supervision size improves performance at the beginning of growth, but then the performance decreases.

Figure 7: Ablation study of kernel size of max pooling on the COCO val set. The optimization can improve inference speed, but the speedup slows down as the kernel size increases.

Because it can bring performance gains without increasing overhead in inference, we recommend supervising all features.

Resolution of HR supervision. The higher resolution of the heatmap results in lower quantization error, which also applies to intermediate supervision. High-resolution heatmaps can help the model to supervise the merged features well. However, with the common issue of high-resolution heatmaps, there is a squared increase in the number of heatmaps, e.g., for a feature $y \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C}$, if we use a ground truth $G \in \mathbb{R}^{H \times W}$ to perform supervision, then 1024 heatmaps $h \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32}}$ have to be predicted for each keypoint. This not only has to learn a large number of parameters but also is not conducive to the supervision of intermediate features. To balance them, we set the resolution to $s \in \{\frac{H}{8} \times \frac{W}{8}, \frac{H}{4} \times \frac{W}{4}, \frac{H}{2} \times \frac{W}{2}, H \times W\}$ for our experiments. According to Fig. 6, we can see that the performance of the model improves when the resolution grows to $\frac{H}{4} \times \frac{W}{4}$, after which, on the contrary, it decreases. Therefore, the recommended settings is $\frac{H}{4} \times \frac{W}{4}$.

Kernel size of max pooling. We downsample heatmaps by max pooling to mitigate the effect of the over-resolution of heatmaps. To verify whether our optimization can improve the inference speed, we set the kernel size $ks = \{1, 2, 4, 8, 16, 32\}$ and conduct experiments at two input resolutions. Note that $ks = 1$ means not to use our proposed

Table 5: Ablation study of different heads on the COCO val set. We evaluate all heatmaps of the output. The SR head is significantly better than the Simple head in performance.

Supervision head	Prediction head	Params	AP	AR
Inter head	Simple head	24.71M	69.8	75.7
SR head	Simple head	24.71M	70.0	75.8
Inter head	SR head	24.80M	72.8	78.3
SR head	SR head	24.80M	**73.3**	**78.8**

Table 6: Ablation study of kernel size of SR heads on the COCO val set. As the kernel size increases, the performance is gradually improved.

Kernel size				Params	AP	AR
M_5	M_4	M_3	M_2			
1	1	1	1	24.71M	72.9	78.4
3	3	3	3	24.72M	73.1	78.6
3	5	7	9	24.80M	**73.3**	**78.8**

optimization but to decode the high-resolution heatmap directly. As shown in Fig. 7, our proposed optimization can improve inference speed. However, as the kernel size keeps increasing, the speedup slows down, and when the kernel size reaches 16, the inference speedup almost stops. Therefore, we set the kernel size to 16 in the inference process.

Heads for supervision and prediction. There are previous methods (*e.g.*, RSN [3] , MSPN [19]) to supervise the features at different scales. However, they first generate heatmaps for each keypoint by 1×1 convolution and then refine heatmaps by multiplying 3×3 convolution. Finally, the heatmaps are scaled to the desired size by interpolation with corner alignment, which may result in feature misalignment. We refer to this supervision head as the "Inter" head. To verify the performance of the SR head in intermediate supervision and prediction, we performed with different heads as supervision and prediction heads in our experiments. The result (Table 5) indicates that the SR head outperforms Inter head in supervision and outperforms Simple Head in prediction.

Kernel size of SR heads. We first use the convolution of 1×1 kernel to convert the feature map into keypoint embeddings. Then we use the grouped convolution intended to achieve a larger perceptual field with a smaller number of parameters. To verify the necessity of a sizeable perceptual field, we set different kernel sizes of convolution for the experiment. From Table 6, it can be concluded that the convolution of large kernel size can achieve better performance. Therefore, we set the SR heads with increasing kernel size. Besides, as can be seen from the table, the number of parameters does not improve much as the size of the convolution kernel increases due to the small number of channels at each keypoint embedding.

4.3 MPII Human Pose Estimation

Dataset & Evaluation metric. We have validated the effectiveness of our proposed method on the COCO dataset, and for further validation, we conduct experiments on the MPII dataset [23]. The MPII dataset contains about 25000 images. Each image contains at least 1 person. There are about 40000 individuals labeled with 16 keypoints

Table 7: Comparison with top-down methods on MPII validation set. SRPose can achieve significant performance improvement compared to baselines on different backbones. Reg: regression-based approach; HM: heatmap-based approach.

Method	Backbone	Type	PCKh@0.5
RLE [16]	ResNet-50	Reg.	85.8
Integral [37]	ResNet-101	Reg.	87.3
PRTR [18]	HRNet-W32	Reg.	89.5
SimBa [45]	ResNet-50	HM.	88.2
HRNet [35]	HRNet-W32	HM.	90.1
SimCC [20]	HRNet-W32	HM.	90.0
TokenPose [21]	L/D24	HM.	90.2
SRPose	ResNet-50	HM.	89.1
SRPose	HRNet-W32	HM.	**90.5**

Table 8: Comparison with top-down methods on CrowdPose test set. SRPose can improve the performance of different baselines in dense pose scenes.

Method	Backbone	AP	AP^E	AP^M	AP^H
SimBa [45]	ResNet-50	63.7	73.9	65.0	50.6
HRNet-W32 [35]	HRNet-W32	66.4	74.0	67.4	55.6
SimCC [20]	HRNet-W32	66.7	74.1	67.8	**56.2**
SRPose	ResNet-50	64.7	74.9	65.8	52.3
SRPose	HRNet-W32	**67.8**	**77.5**	**69.1**	55.6

in the dataset, which is divided into 28000 for training and the rest for testing. The head-normalized probability of correct keypoints (PCKh) is used to evaluate the performance.

Results on the validation set. The results of the MPII validation set are shown in Table 7. We used Res50 and HRNet-W32 as the backbone for our experiments, which improved by 0.9 and 0.5 in PCKh compared to SimpleBaseline-Res50 and HRNet-W32, respectively.

4.4 CrowdPose

Dataset & Evaluation metric. To verify the performance of SR-Pose in dense pose scenes, we further experiment on the Crowd-Pose dataset [17]. The CrowdPose dataset has more crowded scenes, which is the most distinctive feature distinguished from other datasets. It contains a total of 20K images and 80K human instances. The images are divided into 10000 for training, 2000 for validation, and the rest for testing. In addition to the consistent evaluation metrics with COCO [23], the CrowdPose dataset has metrics AP^E and AP^H, where AP^E is the AP score for relatively simple ones and AP^H is the AP score for hard ones. We follow the original paper to detect human instances by YoloV3 [32] and then test performance on the test set of CrowdPose.

Results on the test set. The results of the CrowdPose test set are shown in Table 8. From the experimental results, it can be seen that our proposed SRPose is equally effective in the scenario of the dense pose. Compared with the original SimpleBaseline-Res50 and HRNet-W32, our SRPose improves the AP by 1.0 and 1.4, respectively.

5 CONCLUSION

In this paper, we propose a brand new head, SR head, which can predict high-resolution heatmaps (the same resolution as the input) to alleviate the quantization error and prevent the dependency on post-processing with heatmap-based methods. Besides, to reduce the training difficulty for obtaining large-resolution, we propose SRPose that applies the SR head for HR supervision in a coarse-to-fine manner during the training process. Extensive experiments on COCO, MPII, and CrowdPose benchmarks demonstrate that SRPose can enhance the performance of top-down methods, and SR head can also enhance the performance of bottom-up methods.

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2d human pose estimation: new benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 3686–3693.

[2] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. 2011. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*. Springer, 29–39.

[3] Yuanhao Cai et al. 2020. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*. Springer, 455–472.

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43, 1, 172–186.

[5] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. 2016. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4733–4742.

[6] Haoming Chen, Runyang Feng, Sifan Wu, Hao Xu, Fengcheng Zhou, and Zhenguang Liu. 2022. 2d human pose estimation: a survey. *arXiv preprint arXiv:2204.07370*.

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7103–7112.

[8] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. 2018. Revisiting rcnn: on awakening the classification power of faster rcnn. In *Proceedings of the European conference on computer vision (ECCV)*, 453–468.

[9] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. 2020. Higherhrnet: scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5386–5395.

[10] MMPose Contributors. 2020. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose. (2020).

[11] Nicola Garau, Niccolò Bisagno, Piotr Bródka, and Nicola Conci. 2021. Deca: deep viewpoint-equivariant human pose estimation using capsule autoencoders. *arXiv preprint arXiv:2108.08557*.

[12] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. 2018. Detect-and-track: efficient pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 350–359.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[14] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. 2020. The devil is in the details: delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5700–5709.

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

[16] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. 2021. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11025–11034.

[17] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. 2019. Crowdpose: efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10863–10872.

[18] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1944–1953.

[19] Wenbo Li et al. 2019. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*.

[20] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. 2022. Simcc: a simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision*. Springer, 89–106.

[21] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. 2021. Tokenpose: learning keypoint tokens for human pose estimation. *arXiv preprint arXiv:2104.03516*.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: common objects in context. In *European conference on computer vision*. Springer, 740–755.

[24] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[25] Sahar Rahimi Malakshan, Mohammad Saeed Ebrahimi Saadabadi, Moktari Mostofa, Sobhan Soleymani, and Nasser M Nasrabadi. 2023. Joint super-resolution and head pose estimation for extreme low-resolution faces. *IEEE Access*, 11, 11238–11253.

[26] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. 2021. Tfpose: direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*.

[27] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. 2022. Poseur: direct human pose regression with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*. Springer, 72–88.

[28] Alejandro Newell, Zhiao Huang, and Jia Deng. 2017. Associative embedding: end-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*.

[29] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

[30] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. 2019. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6951–6960.

[31] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. Personlab: person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 269–286.

[32] Joseph Redmon and Ali Farhadi. 2018. Yolov3: an incremental improvement. *arXiv preprint arXiv:1804.02767*.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[34] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883.

[35] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5693–5703.

[36] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. 2017. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 2602–2611.

[37] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. 2018. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision*, 529–545.

[38] Zhi Tian, Hao Chen, and Chunhua Shen. 2019. Directpose: direct end-to-end multi-person pose estimation. *arXiv preprint arXiv:1911.07451*.

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

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[41] Heng Wang and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, 3551–3558.

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

[43] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. 2021. Probabilistic monocular 3d human pose estimation with normalizing flows. *arXiv preprint arXiv:2107.13788*.

[44] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. 2020. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*. Springer, 527–544.

[45] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, 466–481.

[46] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. 2022. Vitpose: simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*.

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

[48] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. 2021. Hrformer: high-resolution vision transformer for dense predict. *Advances in Neural Information Processing Systems*, 34, 7281–7293.

[49] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. 2021. Learning skeletal graph neural networks for hard 3d pose estimation. *arXiv preprint arXiv:2108.07181*.

[50] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. 2020. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7093–7102.

[51] Zhizhuo Zhang, Lili Wan, Wanru Xu, and Shenghui Wang. 2021. Estimating a 2d pose from a tiny person image with super-resolution reconstruction. *Computers & Electrical Engineering*, 93, 107192.

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

[53] Shihao Zou, Chuan Guo, Xinxin Zuo, Sen Wang, Pengyu Wang, Xiaoqin Hu, Shoushun Chen, Minglun Gong, and Li Cheng. 2021. Eventhpe: event-based 3d human pose and shape estimation. *arXiv preprint arXiv:2108.06819*.