Lightweight Human Pose Estimation Algorithm Based on Polarized Self-Attention

Liu Shengjie
Beijing Union University
He Ning (xxthening@buu.edu.com)
Beijing Union University
Wang Cheng
Beijing Union University
Yu Haigang
Beijing Union University
Han Wenjing
Beijing Union University

Research Article

Keywords: human pose estimation, polarized self-attention, ghost module, coordinate decoding

Posted Date: April 29th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1599154/v1

Noname manuscript No.
(will be inserted by the editor)

Lightweight Human Pose Estimation Algorithm Based on Polarized Self-Attention

Shengjie Liu · Ning He · Cheng Wang · Haigang Yu · Wenjing Han

Received: date / Accepted: date

Abstract In recent years, human pose estimation has been widely used in human-computer interaction, augmented reality, video surveillance, and many other fields, but the task of pose estimation still faces many challenges. To address the large number of parameters and complicated calculation in the current mainstream human pose estimation network, this paper proposes a lightweight pose estimation network (Lightweight Polarized Network, referred to as LPNet) based on a polarized self-attention mechanism. First, ghost convolution is used to reduce the number of parameters of the feature extraction network; second, by introducing the polarized self-attention module, the pixel-level regression task can be better solved, the lack of extracted features due to the decrease in the number of parameters can be reduced, and the accuracy of the regression of human keypoints can be improved; finally, a new coordinate decoding method is designed to reduce the error in the heatmap decoding process and improve the accuracy of keypoint regression. The method proposed in this paper was evaluated on the human keypoint detection datasets COCO and MPII, and compared with the current mainstream methods. The experimental results show that the proposed method greatly reduces the number of parameters of the model while ensuring a small loss in accuracy.

Keywords human pose estimation · polarized self-attention · ghost module · coordinate decoding

✉Ning He
E-mail: xxthening@buu.edu.com

Shengjie Liu
E-mail: 1290542855@qq.com

Cheng Wang
E-mail: 973862384@qq.com

Haigang Yu
E-mail: 1184555290@qq.com

Wenjing Han
E-mail: 1479731844@qq.com
1 Beijing Union University, Beijing 100101, China 2 Beijing Information Science and Technology University, Beijing 100101, China

1 Introduction

Human pose estimation is an important research direction in the field of computer vision. Its purpose is to locate the coordinates of human keypoints from video or image data. This task is the preprocessing step of many visual tasks such as pose tracking and human action recognition. Currently, conventional human pose estimation network research is carried out along the direction of deepening the depth of the network, expanding the resolution of the feature map, and designing different resolution networks for multi-scale feature fusion and feature extraction. Such networks need the support of high-performance computing equipment and face many problems, such as a large number of parameters, long training times, and difficulties deploying on low-performance computing equipment, and hence they cannot be implemented in practical applications. Therefore, under the premise of ensuring little loss in keypoint detection accuracy, further reducing the parameters of the model is a problem to be solved in the current human pose estimation task.

Human pose estimation approaches based on deep learning can be divided into top-down and bottom-up methods. The top-down method first performs human object detection on the input image to obtain human objects with bounding boxes. Then, the bounding box is cropped to the size of a single human body, and feature extraction is performed using the pose estimation network to obtain the coordinates of each keypoint of the human body. In 2016, Wei et al. [25] designed the convolutional pose machine network, which uses convolutional layers to express texture information and spatial information, and designed a multi-stage structure to improve the detection performance of single keypoint s. In 2017, Fang et al. [6] designed the regional multi-person pose estimation network, focusing on the problems of detection frame positioning error and repeated detection in the top-down method of target detection algorithms. The human body bounding box is optimized by the spatial transformation network, which overcomes the influence of the target detection algorithm error on the subsequent keypoint detection task. In 2018, Chen et al. [4] designed the CPN (cascaded pyramid network), which mainly focuses on the difficulty of detecting different types of joint points, and designed two two-stage networks, GlobalNet and RefineNet, which further improve the accuracy of detection for more difficult keypoints (occluded keypoints). In 2019, Sun et al. designed a more representative network called the HRNet (high-resolution network) [19], which is characterized by a new parallel multi-resolution fusion architecture that can better extract high-resolution features and improve the detection performance for small and medium-sized people. In 2021, Rawal et al. [11] designed the MIPNet network structure to better cope with the crowding problem in the pose estimation task.

The bottom-up method first performs global keypoint detection on the input image to obtain all keypoints in the image. Then, using the positional relationships of human joints, the joint points are combined into multiple groups of independent human keypoints using a clustering algorithm. In 2017, Cao et al. [2] proposed Openpose and designed a classic keypoint clustering algorithm called part affinity fields that can simultaneously encode the position and direction of joint points to balance keypoint detection speed and accuracy. In 2018, George et al. [18] proposed PersonLab, which uses the combination of a heatmap and offset to predict the position of joint points, which better solves the problem of mutual occlusion between joint points. In 2020, Cheng et al. [5] designed HigherHRNet, which is

an improved version of HRNet, and applied it to the bottom-up approach. There is a performance gap between bottom-up and top-down methods, and hence in 2021, Geng et al. [7] applied adaptive convolution to the keypoint regression part of the pose estimation task, which further advanced the performance of bottom-up methods.

Designing a lightweight network not only includes the exploration of network structure design, but also the application of model compression technologies such as knowledge distillation and model pruning. Lightweight network design further facilitates the application of deep learning technology in mobile terminals and embedded devices. Lightweight network design refers to further reducing the amount and complexity of model parameters while maintaining model accuracy. MobileNet, proposed by Google in 2017 [9], was the first convolutional neural network that is small in size, low in computational complexity, and suitable for mobile devices. The network mainly relies on deep separable convolution and reasonable structural design to realize network parameters. In the same year, Zhang et al. proposed the ShuffleNet [30], which mainly adopts point-by-point convolution and a channel shuffling structure that greatly reduces the number of calculations of the model while ensuring its accuracy. In 2020, Kai et al. [8] designed a new network called GhostNet, that can overcome the need for extra parameters caused by convolution, further improving the model speed and reducing the amount of computation. In 2021, Yu et al. [28] designed Lite-HRNet, which integrates ShuffleNet into a high-resolution network and reduces the computational complexity while improving performance. Network models such as MobileNet and ShuffleNet are designed to make the model smaller and faster by employing a more efficient network structure rather than compressing or migrating a large trained model. The advantage of this method is that it can be better applied in in actual vision tasks.

In summary, the aim of this research is to design an efficient and lightweight human pose estimation network that can reduce the parameters of the network while maintaining high detection accuracy. The feature extraction network adopts a high-resolution network and improves it. Combining the polarized self-attention (PSA) mechanism and the ghost network structure, a lightweight PSA module is designed to replace the basic module in the high-resolution network, which reduces the number of parameters and retains important spatial and channel information to ensure the accuracy of the model. An unbiased coordinate decoding method is also proposed in this paper. After the predicted heatmap is obtained from the feature extraction network, accurate coordinate decoding is performed on the predicted joint points of the heatmap, and the joint point coordinates of the final regression are further refined to improve the detection accuracy of the joint points. Finally, experiments were performed on two mainstream datasets, MPII and COCO, to verify the effectiveness of the designed network.

2 Related work

2.1 High-Resolution Network

Because high-resolution networks can maintain high-resolution representations throughout the network, they are widely used in pixel-level regression tasks, such

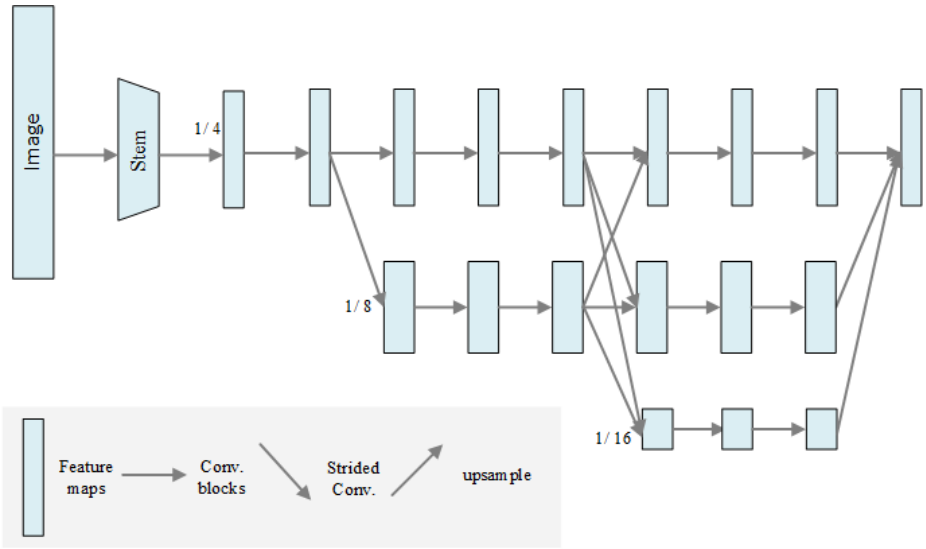

Fig. 1 Simplified HRNet structure.The network structure mainly contains features of four different resolutions, which are 1/4,1/8,1/16,1/32,and features are transferred between features of different resolutions. The low-resolution features are transferred by upsampling, the high-resolution features are transferred by strided convolution, and the features of different resolutions are fused in the final stage

as semantic segmentation, human pose estimation, and many other visual tasks, and have achieved remarkable results. Most of the current pose estimation tasks use high-resolution networks as the backbone network, and the pose estimation networks proposed in the past two years such as HigherHRNet [5] and DEKR [7] have also been designed and improved based on this network. A high-resolution network can improve the extraction of local joint point information, and hence the commonly used high-resolution network HRNet was selected as the basic network for the proposed method. Its structure is different from the traditional concatenated structure. The feature information of different resolutions cannot be fused in the form of a connection, resulting in poor joint point regression results. HRNet uses a parallel method to realize the fusion of information between feature maps of different resolutions and realizes the fusion of multi-scale features through multiple cross-parallel convolutions to enhance the high-resolution feature information so that the entire network can maintain a high-resolution representation. This improves the accuracy of joint point regression for human pose estimation tasks. A brief overview of the HRNet network structure is shown in Fig. 1.

2.2 Ghost Module

To meet the requirements for a lightweight model for the human pose estimation task, the proposed method uses ghost convolution as the main approach to reduce the weight of the network. Because deep convolutional neural networks usually consist of a large number of convolutions, this leads to a large amount of computational overhead. Although recent approaches such as MobileNet and ShuffleNet

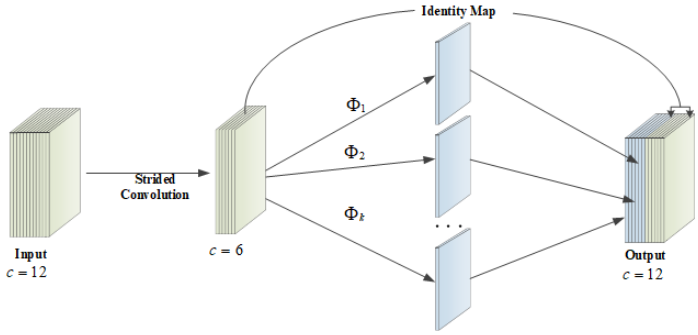

Fig. 2 Ghost convolution module

have introduced depthwise convolution or shuffling (channel shuffle) operations to build efficient convolutional neural networks with smaller convolutional filters (number of floating point operations), the structure of the 1×1 convolution kernels still occupies a considerable amount of memory and FLOPs.

Like the above two recently proposed convolutional networks , ghost convolution is divided into two steps. First, the normal convolution calculation is used to obtain a real feature map with a small number of channels, and then a cheap operation is used to pass the real features through a linear transform. A similar feature map is obtained by the transformation. Hence, the real feature map is identically mapped and the similar feature map is spliced to form a new output. The ghost convolution module is shown in Fig. 2.

In the specific calculation, it is assumed that the input is $X \in \Re^{c \times h \times w}$, where c is the number of input channels, and h and w are the height and width of the input data, respectively. The operation of generating the feature map by the convolutional layer is as follows:

$$Y = X * f + b \tag{1}$$

where $*$ represents the convolution operation, b is the bias term, $Y \in \Re^{h' \times w' \times n}$ represents the output feature map of the n dimensional channel, and $f \in \Re^{c \times k \times k \times n}$ represents the convolution filter of this layer. In addition, h' and w' are the height and width of the output data, respectively, and $k \times k$ is the kernel size of filter f. For the convolution operation of the general process, the number of floating-point operations per second can be calculated by $n \cdot h' \cdot w' \cdot c \cdot k \cdot k$, because the number of filters n and the number of channels c are very large, so the calculation results are usually in the thousands.

As shown in Equation (1), the number of parameters to be optimized (in f and b) is determined by the dimensions of the input and output feature maps. There will be redundant feature maps in the output of ordinary convolutional layers, and some feature maps will be very similar. The process of generating such feature maps will waste a lot of computation. If this type of feature map is obtained by linear transformation from part of the real feature map, the amount of calculation will be significantly reduced. Moreover, such raw features are usually small and produced by ordinary convolution. Specifically, m original feature maps

$Y' \in \Re^{h' \times w' \times m}$ are generated by one convolution, as follows:

$$Y' = X * f' \tag{2}$$

The filter in Equation (2) is expressed as $f' \in \Re^{c \times k \times k \times m}$, where m is less than the number of convolution kernels n. Other hyperparameters are consistent with ordinary convolution to ensure the size of the output feature map. To obtain the required n feature maps, an inexpensive linear transform is applied to the original features in Y', resulting in s phantom features. The specific calculation process is as follows:

$$y_{ij} = \Phi_{i,j}\left(y'_i\right), \forall i = 1, ..., m, j = 1, ..., s, \tag{3}$$

where y'_i is the i-th original feature map in Y'. In the above function, $\Phi_{i,j}$ represents the j-th linear operation, which is used to generate the j-th phantom feature map y_{ij}, indicating that y_{ij} can have one or more phantom feature maps $\{y_{ij}\}_{j=1}^{s}$. The role of $\Phi_{i,s}$ is to preserve the identity mapping of the original feature map. Using an inexpensive linear operation, $n = m \cdot s$ feature maps $Y = [y_{11}, y_{12}, ..., y_{ms}]$ can be obtained as the output data of the ghost model. Linear operation Φ acts on each channel, and the amount of calculation is much lower than that of an ordinary convolution operation.

In terms of computational complexity, the ghost module has an identity map and $m \cdot (s-1) = \frac{n}{s} \cdot (s-1)$ linear operations, and the average convolution kernel size in each linear operation is $d \times d$. Ideally, the $n \cdot (s-1)$ linear operations can have different shapes and parameters, but limited by the CPU and GPU, online inference will be hindered. The theoretical acceleration of using ghost convolution and using ordinary convolution is expressed as follows:

$$r_s = \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{4}$$

In Equation (4), the magnitude of $d \times d$ is similar to that of $k \times k$ and $s \ll c$. The same parameter compression ratio calculation can be expressed as follows:

$$r_c = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{5}$$

According to Equations (4) and (5), the parameter compression ratio is approximately equal to the speedup ratio. Ghost convolution can be easily embedded into other network models in a plug-and-play manner, but some extracted features may be lost when reducing the number of parameters and amount of computation. Therefore, the use of ghost convolution should be considered with respect to specific requirements, and reducing the number of parameters cannot be blindly pursued while ignoring the performance of the model.

2.3 Attention Mechanisms

In recent years, attention mechanisms [21, 22] have been widely used in various computer vision tasks. The main function of an attention mechanism is to improve the feature extraction network's ability to extract pixel information in pixel-level regression tasks, overcome the loss of spatial information in traditional convolution

operations, and achieve better regression results for subtle joints in pose estimation tasks.

Attention mechanisms can be roughly divided into two categories: strong attention and soft attention mechanisms. Because strong attention is a random prediction that emphasizes dynamic changes, although its performance is good, its application is very limited because of its non-differentiable nature. On the contrary, soft attention is differentiable everywhere, that is, it can be obtained by neural network training based on the gradient descent method, so its application is relatively wide. A soft attention mechanism is divided according to the different dimensions of attention. The current mainstream attention mechanism can be divided into the following three types: channel attention, spatial attention, and self-attention. For channel attention, the purpose is to model the correlation between different channels (feature maps), automatically obtain the importance of each feature channel through network learning, and finally assign different weights to each channel. Weight coefficients are used to strengthen important features and suppress non-important features. Representative methods include SENet [10] and ECANet [23]. For spatial attention, the purpose is to improve the feature expression of key regions. In essence, the spatial information in the original image is transformed into another space and the key information is retained through the spatial transformation module. A weight mask is then generated for each position and weighted output, thereby enhancing the specific target regions of interest while weakening the irrelevant background regions. Representative methods include CBAM [26] and A^2Net [3].

Self-attention is a variant of the attention mechanism whose purpose is to reduce the dependence on external information and use the inherent information inside the feature to interact with the attention as much as possible. In the self-attention mechanism, each input tensor is used to compute an attention tensor, which is then reweighted by that attention tensor. Following its success in sequence modeling and generative modeling tasks, self-attention has become a standard component for capturing long-range interactions. Representative methods include NLNet [24], GCNet [1], and SCNet [15].

3 Proposed method

The design of the a lightweight PSA pose estimation network (LPNet) in this paper starts from the problems of the large number of parameters in mainstream network and the difficulty of real-time detection in the pose estimation task. A lightweight network can effectively reduce the number of network parameters while improving the network's ability to extract features from the two dimensions of channel and space and reducing the error in the process of heatmap regression to keypoints. In the human pose estimation task, an optimal balance between parameter quantity and accuracy is achieved. A lightweight ghost module is introduced and embedded into the feature extraction network to reduce the number of parameters and amount of computation of the network. The introduction of the PSA module improves the network's ability to extract channel and spatial features and ensure the accuracy of heatmap prediction on the premise of a small increase in parameters. A new coordinate decoding method is proposed to reduce the error in the process of heatmap regression to keypoints..

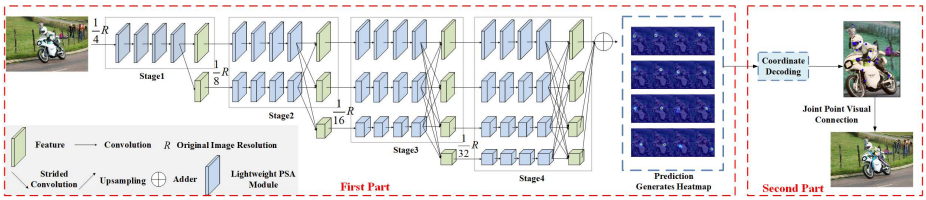

Fig. 3 LPNet network structure.The feature extraction part adopts the improved HRNet, the blue module represents the Lightweight PSA Module.17 keypoint heatmaps are obtained by extracting features for prediction, and specific key point coordinates are obtained through coordinate decoding

3.1 LPNet

Because of the particular characteristics of pixel-level regression tasks, high-resolution networks perform better on pixel-level regression tasks. Therefore, the design presented in this paper is based on a high-resolution network. The designed LPNet network is divided into two main parts (Fig. 3). The first part of the network extracts the features of the input image and predicts the generation of joint hot spots. The second part decodes the coordinates of the predicted heatmap to obtain the joint point coordinates. The first part mainly improves the feature extraction part, introduces the lightweight method into the feature extraction network, and uses the designed lightweight PSA module to replace the basic modules in the four stages to reduce the number of feature extraction network parameters. Moreover, from the channels, it learns finer pixel-level information in the spatial dimension, overcomes some shortcomings in traditional convolutional networks, and ensures the efficiency of the feature extraction process. The second part mainly consists of a new coordinate decoding method to overcome the error of the traditional heatmap coordinate decoding process and improve the accuracy of the heatmap decoding joint point coordinates.

3.2 Self-Attention for Pixel-wise Regression

In the pixel-level regression task, a deep convolutional network mainly learns the weighted feature map from two types of information: i) the classification of the pixel from the perspective of the channel and ii) the detected pixel locations belonging to the same semantics from the perspective of space. Determining these two types of information is the main purpose of the current attention mechanism and they embody its advantages and disadvantages. However, none of the current existing self-attention methods can achieve a specific weighting of channels and spaces, which is a challenge in the design of attention mechanisms.

3.2.1 PSA Module

To solve the problem of computational complexity and memory explosion if the dimension reduction is not performed when modeling channels and spaces simultaneously, the PSA mechanism was proposed. The PSA mechanism adopts the mechanism of polarization filtering, which is similar to the mechanism of an optical

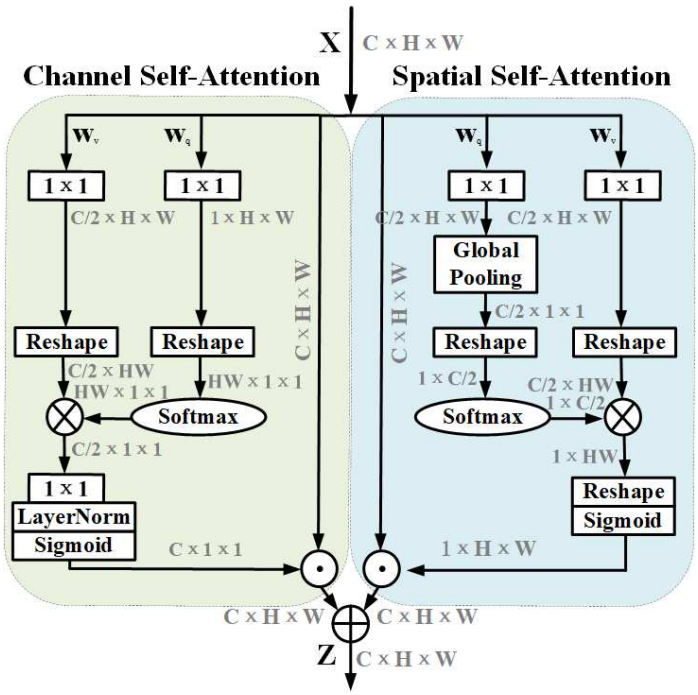

Fig. 4 Structure of the PSA module

lens. During photography, all lateral light is reflected and refracted. Polarization filtering only allow lights orthogonal to the transverse direction to pass through to improve the contrast of imaging. However, during the filtering process, the total intensity will be lost, and hence the filtered light usually has a small dynamic range, so it is necessary to carry out additional amplification to restore the details in the original scene.

The design of the PSA mechanism (Fig. 4) is based on the above ideas. It compresses the current features in one direction and improves the intensity range of the loss, which is divided into the following two main structures: i) the filtering module, which completely collapse the features of one dimension (such as the channel dimension), while keeping the orthogonal dimension (such as the spatial dimension) at a higher resolution, and ii) the HDR (high dynamic range) module, in which the softmax function is used on the smallest features in the attention module to increase the attention range and the sigmoid function is used for dynamic mapping.

As shown in Fig. 4, the PSA module is divided into two branches, the channel branch and the spatial branch. When the input only passes through the channel branch, the weight of the channel branch is expressed as $A^{ch}(X) \in \Re^{C\times 1\times 1}$, and the calculation process is as follows:

$$A^{ch}(X) = F_{SG}[W_{Z|\theta_1}((\sigma_1(W_v(X)) \times F_{SM}(\sigma_2(W_q(X))))] \tag{6}$$

Here,W_q ,W_v , and W_z are 1×1 convolutions; σ_1 and σ_1 represent two-dimensional changes;$F_{SM}(\cdot)$ is the softmax function; "$\times$" represents the matrix dot-product

operation; $F_{SM}(X) = \sum_{j=1}^{N_p} \frac{e^x j}{\sum_{m=1}^{N_p} e^{xm}} x_j$; and the number of channels between $W_v|W_q$ and W_z is $C/2$. The output of the channel dimension-only branch is $Z^{ch} = A^{ch}(X)\odot^{ch} X \in \Re^{C\times H\times W}$, where $\odot^{ch}$ is the channel multiplication operator. When the input only passes through the channel branch, 1×1 convolution is used to convert the input feature X into Q and V, where the channel of Q is completely compressed and the channel of V retains its higher dimension ($C/2$). Because the channel of Q is compressed, based on the idea of the PSA mechanism, information needs to be converted to HDR, so the softmax function is used to enhance the information of Q. Then, matrix multiplication is performed between Q and V, and 1×1 convolution and LayerNorm are used to restore the channel dimension to C. Finally, the sigmoid function is used to normalize all parameters.

When the input only passes through the spatial branch, the weight of the spatial branch is expressed as $A^{sp}(X) \in \Re^{1\times H\times W}$, and the calculation is as follows:

$$A^{sp}(X) = F_{SG}[\sigma_3((F_{SM}(\sigma_1(F_{GP}(W_q(X)))) \times \sigma_2(W_v(X)))] \tag{7}$$

where W_q and W_v are both standard 1×1 convolutions; θ_2 denotes the intermediate parameters between convolution channels; σ_1,σ_2, and σ_3 represent the three-dimensional changes; and $F_{SM}(\cdot)$ is the softmax function. Furthermore, F_{GP} denotes the global pooling operator, where $F_{GP}(X) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X(:,i,j)$, and "$\times$" means matrix dot product operation. The output of the spatial dimension-only branch is $Z^{sp} = A^{sp}(X)\odot^{sp} X \in \Re^{C\times H\times W}$, where $\odot^{sp}$ is the spatial multiplication operator. When the input only passes through the spatial branch, as in the channel branch, 1×1 convolution is used to convert the input feature X into Q and V, and for feature Q, the spatial dimension is compressed by global pooling and converted to a size of 1×1; by contrast, the spatial dimension of feature V remains high ($H\times W$). Because the spatial dimension of Q is compressed, based on the idea of the PSA mechanism, the softmax function is used to enhance the information of Q. Then, matrix multiplication is performed between Q and V, a matrix transform is used to reshape the result, and the sigmoid function is used to normalize all parameters.

The channel and space branches are combined in parallel as follows:

$$PSA_p(X) = Z^{ch} + Z^{sp} = A^{ch}(X)\odot^{ch} X + A^{sp}(X)\odot^{sp} X \tag{8}$$

where + represents the element-wise addition operator. In contrast to other self-attention mechanisms, PSA retains the highest attention resolution in both channel ($C/2$) and space ($[W,H]$), and can capture finer channel-wise and spatial details when processing pixel-level tasks. In addition, in the single-channel branch part, softmax re-weighting as well as squeeze and excitation are adopted, and both SENet(Squeeze-and-Excitation Network) and GCNet(Global Context Network) benefit from this approach. In the single spatial branch part, not only is the full spatial resolution maintained, but more learnable parameters are retained internally for nonlinear softmax reweighting, which is a more powerful structure than existing self-attention mechanisms. Because of these advantages, PSA can achieve the optimal improvements in performance for pixel-level regression tasks.

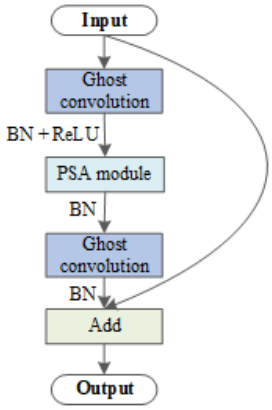

Fig. 5 Lightweight PSA Module

3.3 Lightweight PSA Module

To meet the requirement for a lightweight network for the human pose estimation task, the ghost network and PSA modules were redesigned, and the results is called the lightweight PSA module, as shown in Fig. 5. This module is similar to the BasicBlock module in a high-resolution network, which can extract features and reduce the number of parameters of the overall network. The lightweight PSA module is mainly composed of two ghost convolutions and a PSA module. The first ghost convolution expands the number of channels, and then process the data through normalization and ReLU functions. The processed data are sent to the PSA module to capture finer channel-wise features and spatial features while ensuring high resolution, with almost no increase in the number of parameters and calculations. The data are then normalized again and fed to the next ghost convolution. The second ghost convolution restores the channel to the original number of channels, and finally combines the residual structure principle to sum the data and the data of the feature map to obtain the final output.

3.4 New coordinate decoding method

The ultimate goal of the human pose estimation task is to obtain the coordinate positions of each joint point of the human body in the original image. After predicting the heatmap of human joint points through the pose estimation network, the corresponding resolution recovery is required to convert the results back to the original coordinate space. This conversion process is called coordinate decoding.

The traditional coordinate decoding method is designed according to the specific performance of different models. Specifically, given the heatmap h predicted by the trained model, the peak (m) and sub-peak (s), which is the location of the second largest activation value, are determined. The joint point position prediction is as follows:

$$p = m + 0.25\frac{s-m}{\|s-m\|_2} \tag{9}$$

where $\|\cdot\|_2$ represents the size of the vector. Equation (9) indicates that the joint point position prediction is a shift of 0.25 pixels from the largest activation position to the second largest activation position in the heatmap space. The final coordinate prediction calculation in the original image is as follows:

$$\hat{p} = \lambda p \tag{10}$$

where λ is the resolution reduction ratio.

The main purpose of the pixel shift in Equation (9) is to compensate for the quantization error caused by the downsampling operation. The predicted maximum activation position in the heatmap is not equal to the exact position of each joint point in the original coordinate space. Instead, it is only a rough estimate. Hence, this paper introduces a new decoding strategy.

The new coordinate decoding method mainly focuses on predicting the distribution structure of the heatmap to infer a more accurate maximum activation value position. The specific operation is as follows: To accurately locate the second largest activation, it is assumed that the predicted heatmap follows a two-dimensional Gaussian distribution, just as in the real heatmap. Therefore, the predicted heatmap is expressed as follows:

$$G\left(x;\mu,\sum\right) = \frac{1}{(2\pi)\left|\sum\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)\right) \tag{11}$$

where x represents the pixel location in the predicted heatmap and μ is the Gaussian mean (center) corresponding to the joint location to be estimated. The covariance $\sum$ is a diagonal matrix, expressed as follows, which is consistent with the coordinate encoding process:

$$\sum = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{12}$$

where σ is the standard deviation in both directions. A logarithmic transform is performed on Equation (11), and then the derivative is taken. The specific process is as follows:

$$P\left(x;\mu,\sum\right) = \ln(G) = -\ln(2\pi) - \frac{1}{2}\ln\left(\left|\sum\right|\right) - \frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu) \tag{13}$$

The ultimate goal is to estimate μ. Assuming it is an extreme point in the distribution, the first derivative at position μ should satisfy the following:

$$D'(x)\big|_{x=\mu} = \left.\frac{\partial P^T}{\partial x}\right|_{x=\mu} = \left.-\sum^{-1}(x-\mu)\right|_{x=\mu} = 0 \tag{14}$$

To continue analyzing this situation, Taylor's theorem is used. Activation $p(\mu)$ is approximated using a Taylor series (up to the quadratic term), which evaluates to the following equation at the maximum activation m of the predicted heatmap:

$$P(\mu) = P(m) + D'(m)(\mu - m) + \frac{1}{2}(\mu - m)^T D''(m)(\mu - m) \tag{15}$$

Here, $D''(m)$ represents the second derivative of P evaluated at m, and its specific form is defined as follows:

$$D'(m) = D'(x)\big|_{x=m} = -\sum^{-1} \tag{16}$$

The use of m was chosen to approximate μ because m represents the optimal joint prediction close to μ. Next,$P(\mu)$ and $P(m)$ in Equation (15) are both represented using Gaussian distributions, the constant term is reduced, and we merge Equations (14) to (16), which yields

$$\mu = m - \left(D''(m)\right)^{-1} D'(m) \tag{17}$$

Here, $D''(m)$ and $D'(m)$ can be effectively estimated from the heatmap. As long as μ is available, the coordinates in the original image space can be predicted from Equation (10).

In contrast to standard methods that only consider the second largest activation in the heatmap, the new coordinate decoding fully explores the heatmap distribution statistics to reveal potential maxima more accurately. Theoretically, the method is based on the principled distribution approximation under the assumption of consistent training supervision, that is, the heatmap is a Gaussian distribution. Hence, this method is very computationally efficient and only needs to compute the first and second derivatives at one location in each heatmap. Therefore, the method can be easily integrated into the existing heatmap-based human pose estimation tasks, which will further reduce the error in the heatmap decoding process without increasing the number of parameter calculations.

4 Experimental Results and Analysis

The environment of the experiments reported in this paper was an Ubuntu 18.04.6 LTS 64-bit operating system running on a computer equipped with an Intel(R) Xeon(R) Silver 4216 CPU @2.10GHz; 188.6GiB RAM; GPU RTX3090; and a CUDA v11.0.207, cuDNN v8.2, PyTorch v1.8.0, and Python v3.6.13 software platform. The pre-trained network parameters were taken from a model trained on the ImageNet dataset. In the experiment, the optimizer used the Adam optimizer, the initial learning rate of the model was set to 0.001, and the learning rate decay coefficient was 0.1. The learning rate was decayed after 170 and 200 epochs, respectively, with decay rates of 10-4 and 10-5. The training process ended after 210 epochs.

Datasets: The MPII dataset is a mainstream human pose estimation dataset with single/multiple data types. The dataset includes 25,000 annotated images of more than 40K people, and the image sources are all from YouTube videos. The test set also includes annotations of data such as body part occlusion, three-dimensional torsos, and head orientation.

The COCO dataset is a large, rich dataset for object detection, segmentation, and captioning. This dataset targets environment perception and was mainly collected from complex daily scenes. The target in the image is calibrated by precise segmentation. The images include 91 classes of objects, 328,000 images and 2,500,000 labels. By far the largest dataset for semantic segmentation, it includes 80 categories and consists of more than 330,000 images, 200,000 of which are labeled. The number of individuals in the entire dataset exceeds 1.5 million.

4.1 Evaluation Indicators

The MPII dataset uses the PCK (percentage of correct keypoints) index to evaluate experimental performance. PCK is defined as the proportion of correctly estimated keypoints, which are keypoints for which the normalized distance between the detected keypoints and their corresponding real labels is less than a set threshold. The PCK is calculated as follows:

$$PCK_i^k = \frac{\sum_p \delta\left(\frac{d_{pi}}{d_p^{def}} \leq T_k\right)}{\sum_p 1} \tag{18}$$

$$\mu = m - \left(D''(m)\right)^{-1} D'(m) \tag{19}$$

Here, i represents the i-th keypoint, k represents the k-th threshold T_k, and p represents the p-th pedestrian. Furthermore,d_{pi} represents the Euclidean distance between the predicted value of the i-th keypoint i in the p-th person and the manually labeled value, and d_p^{def} represents the scale factor of the p-th person. The methods for calculating this factor differ in different public datasets. The MPII dataset uses the head diameter of the current person as the scale factor, that is, the upper left point LT of the head and the lower right point RB. Threshold T_k is the manually set threshold,$T_k \in [0 : 0.01 : 0.1]$, PCK_i^k represents the PCK index of the i-th keypoint under threshold T_k, and PCK_{mean}^k represents the mean PCK index for the algorithm under threshold T_k.

The experimental evaluation index of the COCO dataset is the OKS(Object Keypoint Similarity). The equation for OKS is as follows:

$$\frac{\sum_j \exp\left(-d_j^2/2s^2k_j^2\right)\delta\left(v_j > 0\right)}{\sum_j \delta\left(v_j > 0\right)} \tag{20}$$

where d_j is the Euclidean distance between the detected keypoint coordinates and the real value, v_j indicates whether the keypoints of the human body can be observed, s is the size of the detection target, and k_j is the attenuation coefficient of each keypoint.

The OKS is used in the experiment to determine the AP^{50} (the average precision when the IoU is equal to 0.5), AP^{75} (the average precision when the IoU is equal to 0.75). Furthermore, mAP (mean average precision) is the average AP for each category, AP^M is the average precision for a medium-scale human body, AP^L is the average precision for a large-scale human body.

4.2 Analysis of the Results

The LPNet algorithm proposed in this paper is compared with other advanced pose estimation algorithms proposed in recent years. Table 1 shows the results on the MPII validation set. The LPNet algorithm uses approximately a quarter of the parameters used by the baseline network HRNet, but achieves a 0.5 percentage point improvement in accuracy. Compared with other recent attitude estimation methods, LPNet is better in parameter quantity and accuracy. Example results from the MPII validation set are shown in Fig. 6.

Fig. 6 Example results from the MPII validation set

Table 1 Experimental results on MPII validation set. HM: heatmap-based regression, Reg: coordinate-based regression

Method	BackBone	Type	Params	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
SimpleBaseline [27]	ResNet-152	HM	68.6M	97	95.9	90	85	89.2	85.3	81.3	89.6
HRNet [19]	HRNet-W32	HM	28.5M	96.9	96	90.6	85.8	88.7	86.6	82.6	90.1
TokenPose [14]	L/D24	HM	28.1M	97.1	95.9	90.4	86	89.3	87.1	82.5	90.2
Integral [20]	ResNet-101	Reg	45M	-	-	-	-	-	-	-	87.3
PRTR [12]	HRNet-W32	Reg	-	97.3	96	90.6	84.5	89.7	85.5	79	89.5
Poseur [16]	HRNet-W32	Reg	-	-	-	-	-	-	-	-	90.5
Ours	**LPNet**	**HM**	**7.5M**	**97.3**	**96**	**90.9**	**86.8**	**89.2**	**87.5**	**83.1**	**90.6**

Table 2 shows the experimental results on the COCO val2017 dataset. The results show that when the input resolution is 256×192, the AP value of LPNet is 74.0, which is only 0.4% worse than the baseline network HRNet, but the number of network parameters is not as high as that of the baseline network. By increasing the input image scale and the number of input channels, the detection accuracy can be further improved. When the input resolution is 384×288 and the number of channels is 48, the best performance is achieved, yielding an AP of 76.4. Moreover, the number of parameters is still lower than the baseline network with 32 channels. Compared with other classical pose estimation methods, it performs better with respect to parameter quantity and average accuracy. In addition, the experimental results of LPNet were visually evaluated, and example results are shown in Fig. 7. The joint points are connected in the images, showing that LPNet is more precise

Table 2 Results of each model on COCO val2017

Method	BackBone	Input Size	Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass [17]	Hourglass	256 × 192	25.1M	14.3	66.9	-	-	-	-	-
CPN [4]	ResNet-50	256 × 192	27.0M	6.2	68.6	-	-	-	-	-
CPN+OHKM [4]	ResNet-50	256 × 192	27.0M	6.2	69.4	-	-	-	-	-
SimpleBaseline [27]	ResNet-50	256 × 192	34.0M	8.9	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline [27]	ResNet-101	256 × 192	53.0M	12.4	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline [27]	ResNet-152	256 × 192	68.6M	15.7	72	89.3	79.8	68.7	78.9	77.8
HRNet-W32 [19]	HRNet-W32	256 × 192	28.5M	7.1	74.4	90.5	81.9	70.8	81	79.8
HRNet-W48 [19]	HRNet-W48	256 × 192	28.5M	7.27	75.1	90.6	82.2	71.5	81.8	80.4
DARK [29]	HRNet-W48	128 × 96	63.6M	3.6	71.9	89.1	79.6	69.2	78	77.9
Our Method W32	**LPNet-W32**	256 × 192	**7.5M**	**1.92**	**74**	**91.4**	**81.3**	**71.2**	**78.5**	**78.2**
Our Method W48	**LPNet-W48**	256 × 192	**19.3M**	**4.21**	**74.8**	**89.8**	**81.3**	**70.6**	**79.7**	**78.9**
SimpleBaseline [27]	ResNet-152	384 × 288	68.6M	35.6	74.3	89.6	81.1	70.5	79.7	79.7
HRNet-W32 [19]	HRNet-W32	384 × 288	28.5M	16	75.8	90.6	82.7	71.9	82.8	81
HRNet-W48 [19]	HRNet-W48	384 × 288	63.6M	32.9	76.3	90.8	82.9	72.3	83.4	81.2
MSPN [13]	4 × Res-50	384 × 288	71.9M	58.7	76.1	93.4	83.8	72.3	81.5	81.6
Our Method W32	**LPNet-W32**	384 × 288	**7.5M**	**3.9**	**75.8**	**93.2**	**84.5**	**73.5**	**81.4**	**80.5**
Our Method W48	**LPNet-W48**	384 × 288	**19.3M**	**11.3**	**76.4**	**92.6**	**83.6**	**72.2**	**81.7**	**81.1**

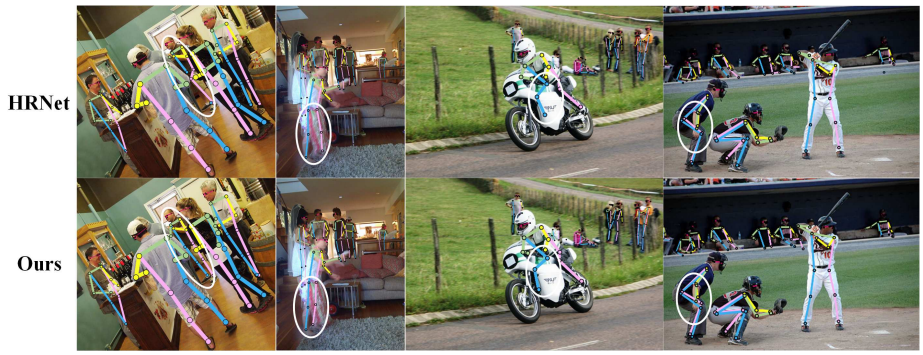

Fig. 7 Example results of LPNet and HRNet on the COCO val2017 dataset

and accurate in estimating joint points than HRNet. Moreover, it performs well in the presence of occlusion.

4.3 Analysis of the Ablation Results

In this study, ablation experiments were performed on the COCO dataset using a high-resolution network with an input channel of 32 as the backbone network and an input image size of 256 × 192. The ablation experiment gradually replaced the basic feature extraction module in the four stages in the high-resolution network with a lightweight PSA module. The experimental results are shown in Table 3. In Table 3, 0 indicates that no basic modules were replaced, and 1–4 indicate that basic modules were replaced in each stage.

As shown in Table 3, as the basic modules in the feature extraction network are replaced stage by stage with the light-weight PSA module in the high-resolution

Table 3 Results of the COCO dataset ablation experiment

Stage Name	Params	GFLOPs	*AP*
0	29.1M	7.10	74.4
1	28.4M	6.64	74.4
1 2	21.6M	5.31	74.2
1 2 3	17.4M	3.6	74.1
1 2 3 4	7.5M	1.92	74.0

network, the number of parameters decreases rapidly and the average precision value decreases. However, thanks to the support of the PSA module, the average accuracy value is still 74.0. The data in the table show that using the lightweight PSA module to replace the basic modules in all four stages can achieve the best performance in terms of network parameter quantity and joint point detection accuracy

Table 4 Results of the COCO dataset ablation experiment

Method	Params	GFLOPs	*AP*
HRNet	29.1M	7.10	74.4
LPNet (Ghost)	5.1M	1.45	72.5
LPNet (Ghost+PSA)	7.5M	1.91	73.6
LPNet (Ghost+PSA+NCD)	7.5M	1.92	74.0

Furthermore, ablation experiments were performed on the PSA module and the new coordinate decoding method. The ghost module, PSA module, and new decoding method were added to the basic network in turn. The results in Table 4 show that only adding the ghost module leads to a sharp reduction in the number of parameters and GFLOPs, but it also causes a large loss in the accuracy of the model. Adding the PSA module improves the accuracy of the network model while increasing the number of parameters by a small amount. The final experimental results show that adding both the ghost and PSA modules as well as using the new coordinate decoding method achieves the best attitude estimation performance.

5 Conclusions

The LPNet proposed in this paper is an improved version of the high-resolution network. The ghost module was combined with the PSA module to create a lightweight PSA module to replace the basic module in the feature extraction network while reducing the number of network parameters and retaining accuracy of the network model. In the final heatmap decoding part, a new coordinate decoding method was introduced that further improves the detection accuracy of keypoints and refines the coordinate positions of the joint points. The advantage of this network is that it has a lightweight architecture, is extremely scalable and easy to use, and provides a new solution and approach to the challenges of current pose estimation tasks such as complex models and a high number of parameters. A large number of experimental results on different datasets demonstrate that the

model has good generalization ability. How to better control the parameters of the network and deploy the model on embedded devices while substantially improving the detection accuracy of the joint points will be the focus of future research.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61872042, 62172045), the Key Project of Beijing Municipal Commission of Education (KZ201911417048), Beijing Union University Talents Strengthening School Selection Program (BPHR2020AZ01, BPHR2020EZ01), National Key Research and Development Program (2018AAA0100804), Beijing Union University Scientific Research Project (No.ZK50202001).

References

1. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0 (2019)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7291–7299 (2017)
3. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Aˆ 2-nets: Double attention networks. Advances in neural information processing systems **31** (2018)
4. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7103–7112 (2018)
5. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5386–5395 (2020)
6. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE international conference on computer vision, pp. 2334–2343 (2017)
7. Geng, Z., Sun, K., Xiao, B., Zhang, Z., Wang, J.: Bottom-up human pose estimation via disentangled keypoint regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14676–14686 (2021)
8. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1580–1589 (2020)
9. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141 (2018)
11. Khirodkar, R., Chari, V., Agrawal, A., Tyagi, A.: Multi-instance pose networks: Rethinking top-down pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3122–3131 (2021)
12. Li, K., Wang, S., Zhang, X., Xu, Y., Xu, W., Tu, Z.: Pose recognition with cascade transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1944–1953 (2021)
13. Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., Sun, J.: Rethinking on multi-stage networks for human pose estimation. arXiv preprint arXiv:1901.00148 (2019)
14. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11313–11322 (2021)
15. Liu, J.J., Hou, Q., Cheng, M.M., Wang, C., Feng, J.: Improving convolutional networks with self-calibrated convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10096–10105 (2020)
16. Mao, W., Ge, Y., Shen, C., Tian, Z., Wang, X., Wang, Z., Hengel, A.v.d.: Poseur: Direct human pose regression with transformers. arXiv preprint arXiv:2201.07412 (2022)

17. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision, pp. 483–499. Springer (2016)
18. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European conference on computer vision (ECCV), pp. 269–286 (2018)
19. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)
20. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European conference on computer vision (ECCV), pp. 529–545 (2018)
21. Tsotsos, J.K.: Analyzing vision at the complexity level. Behavioral and brain sciences **13**(3), 423–445 (1990)
22. Tsotsos, J.K.: A computational perspective on visual attention. MIT Press (2021)
23. Wang, Q., Wu, B., Zhu, P., Li, P., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
24. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7794–7803 (2018)
25. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
26. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp. 3–19 (2018)
27. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV), pp. 466–481 (2018)
28. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., Wang, J.: Lite-hrnet: A lightweight high-resolution network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10440–10450 (2021)
29. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7093–7102 (2020)
30. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848–6856 (2018)