Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.030677

ARTICLE

Lightweight Multi-Resolution Network for Human Pose Estimation

Pengxin Li[1], **Rong Wang**[1,2,*], **Wenjing Zhang**[1], **Yinuo Liu**[1] **and Chenyue Xu**[1]

[1]School of Information and Cyber Security, People Public Security University of China, Beijing, 100038, China

[2]Key Laboratory of Security Prevention Technology and Risk Assessment of Ministry of Public Security, Beijing, 100038, China

*Corresponding Author: Rong Wang. Email: dbdxwangrong@163.com

Received: 18 April 2023 Accepted: 27 July 2023 Published: 15 December 2023

ABSTRACT

Human pose estimation aims to localize the body joints from image or video data. With the development of deep learning, pose estimation has become a hot research topic in the field of computer vision. In recent years, human pose estimation has achieved great success in multiple fields such as animation and sports. However, to obtain accurate positioning results, existing methods may suffer from large model sizes, a high number of parameters, and increased complexity, leading to high computing costs. In this paper, we propose a new lightweight feature encoder to construct a high-resolution network that reduces the number of parameters and lowers the computing cost. We also introduced a semantic enhancement module that improves global feature extraction and network performance by combining channel and spatial dimensions. Furthermore, we propose a dense connected spatial pyramid pooling module to compensate for the decrease in image resolution and information loss in the network. Finally, our method effectively reduces the number of parameters and complexity while ensuring high performance. Extensive experiments show that our method achieves a competitive performance while dramatically reducing the number of parameters, and operational complexity. Specifically, our method can obtain 89.9% AP score on MPII VAL, while the number of parameters and the complexity of operations were reduced by 41% and 36%, respectively.

KEYWORDS

Lightweight; human pose estimation; keypoint detection; high resolution network

1 Introduction

Human pose estimation aims to recognize and localize the body joints in the input image or video. It is now one of the most important tasks in the field of computer vision [1–3]. It has received increasing attention in the last few years and has been used in various applications such as animation [4], sports [5–7], and healthcare [8].

There are two categories in current studies for human pose estimation: heatmap-based methods [9–14] and regression-based methods [15–19]. Heatmap-based methods predict the hotspot mapping corresponding to key points first and then use the heatmap response values to reflect the various locations of the key points on the human body. The regression-based method directly maps the input images to the coordinates of body joints. The predominant approach for pose estimation nowadays is based on heatmaps, which are more precise than regression-based techniques [17]. Therefore, the method proposed in this paper is also heatmap-based.

Although heatmap-based methods are the current mainstream of current human pose estimation methods, they also have limitations. On the one hand, the accuracy of the predicted key points is heavily influenced by the resolution of the heatmap, and there is also the challenge of key point occlusion in real-world scenarios. Consequently, the network architecture of heatmap-based methods is becoming more complex, with a growing number of model parameters required to achieve higher performance, with the computational complexity also rapidly increasing. For example, Simple Baselines [20] adds transposed convolutional layers to ResNet [21] to generate high-resolution representations, Hourglass [22] recovers high-resolution features by stacking multiple hourglass modules, HRNet [23] performs repetitive multiscale fusion by connecting multi-resolution subnets in parallel, and DPIT [24] captures global visual information by integrating both top-down and bottom-up approaches. To achieve higher accuracy, network structures are becoming wider and deeper, resulting in a large number of parameters and increasing FLOPs. As a result, it becomes a major barrier to be deployed in computation-limited devices for pose estimation applications. On the other hand, to address the problem of large parameters and increasing FLOPs, there has been some work on lightweight heatmap-based methods for human pose estimation. Lite-HRNet [25] uses conditional channel weighting instead of pointwise convolution to create a lighter network with fewer parameters and operations. To build efficient CNNs, ShuffleNet [26] and MobileNet [27] introduce shuffle operations or deep convolution. However, these operations come at the cost of loss of precision. Fig. 1 compares different methods in terms of parameters and average accuracy, showing that Hourglass [22], Simple Baselines [20], HRNet [23] and DPIT-B [24] achieve higher accuracy, but require more parameters, while lightweight networks such as Simple Baselines with MobileNetV2 [27], ShuffleNetV2 [26], LiteHRNet18 [25] and LiteHRNet30 [25], though reducing parameters in the network structure, may compromise accuracy. Overall, the existing methods make it difficult to strike a balance between accuracy and the number of parameters.

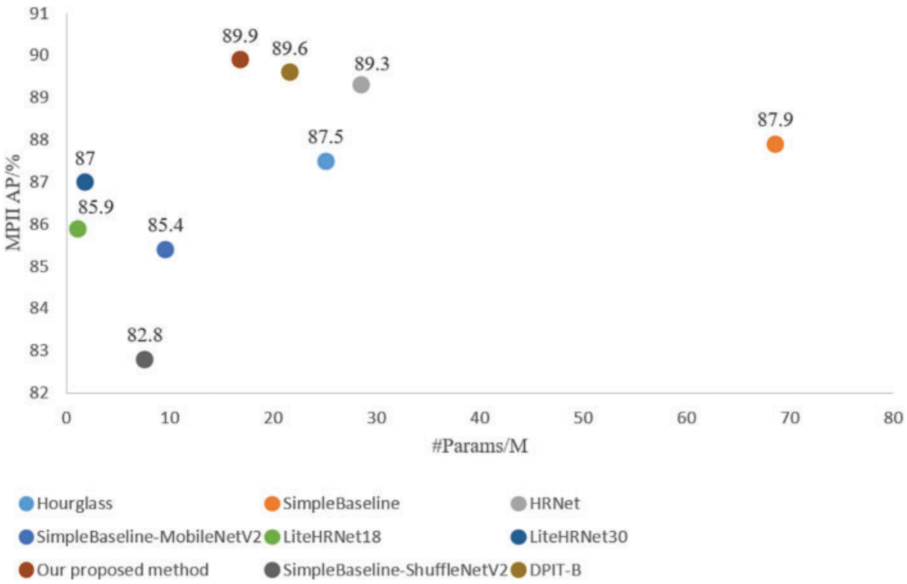

Figure 1: Comparison figure between our method and existing methods. AP score and the number of parameters in the corresponding model are represented by the circular area

Based on these two problems, we have developed a lightweight framework for human pose estimation to achieve a balance between accuracy and the number of parameters. Our approach involves a lightweight feature encoder in the feature extraction stage. The feature encoder employs deep convolution to convolve each channel separately and reassign channel weights, thereby preserving the multi-resolution parallel structure of the network while also reducing the number of model parameters and the complexity of operations. Meanwhile, we add a semantic enhancement module to the deep network to connect the channel semantic information with the spatial semantic information, which eventually enhances global connectivity and improves the accuracy of the network. Furthermore, we introduce a spatial pyramid pooling module to densely connect the dilated convolution and fuse the low-order features. This compensates for the degradation of image resolution and information loss that can arise from downsampling in our lightweight structure. Our main contributions include:

- A lightweight feature encoder (SCR) is proposed to construct a lightweight multi-resolution network for human pose estimation in order to reduce parameters and network complexity while maintaining high performance.

- A semantic enhancement module (SEM) and a spatial pyramid pooling module (DASP) is constructed to compensate for the information loss caused by downsampling and convolution in the lightweight structure and improve the lightweight network performance.

- The experiments show the effectiveness of our method on MPII [28] and COCO [29] datasets, achieving high accuracy and greatly reducing the number of parameters and complexity of the network.

2 Related Works

2.1 Pose Estimation

Human pose estimation has been an active research component. Before the rise of deep learning, in traditional methods of human pose estimation [30], a priori geometrical knowledge is mainly utilized to put the human body structure represented by a template, and then the human pose is detected by a template matching algorithm and constructed. The development of deep learning has promoted the development of various human research [31–33]. Deep learning-based human pose estimation technology has been developed rapidly since 2013. Simple Baselines [20] extracts feature with ResNet [21] as the foundation, then adds three deconvolution layers with batch normalization and ReLU in the final convolution stage to generate high-resolution feature maps and heatmaps. Hourglass network [22] first downscales the image from high to low resolution, then extracts stronger semantic features, upscales the image from low to high resolution, strengthens localization features, and finally generates a high-resolution heatmap. DPIT [24] achieves complementarity by integrating both top-down and bottom-up approaches to capture visual cues from different receptive fields. HRNet [23] recovers the high resolution from the sub-low network's resolution and maintains it throughout the process. It gradually adds branches with lower resolution in parallel, fuses and exchanges information from all branches, and finally generates key point estimates.

Human pose estimation requires the extraction of rich semantic information using deep convolutional neural networks. Traditional methods for obtaining rich semantic information typically use either a high-to-low process or a low-to-high process. HRNet [23] employs a parallel multi-resolution convolutional approach that preserves high-resolution representations and enhances high-resolution feature information through multi-resolution feature map fusion. The network comprises four stages, with the first stage consisting of a high-resolution subnet. In subsequent stages, each stage has one additional low-resolution sub-network compared to the previous stage, with resolutions of 1/2, 1/4, and

1/8 of the first stage's high-resolution sub-network, respectively. The multi-resolution sub-networks are connected in parallel, and the fusion module fuses the information of different resolutions to enhance the network's feature extraction performance, ultimately improving the detection effect of the key point localization for pose estimation.

2.2 Resolution Feature Representations

One of the major challenges in using CNNs for pose estimation and semantic segmentation is the substantial reduction in resolution that occurs due to pooling. To address this issue, in Fully Convolutional Networks (FCN) [34], upsampling is deployed on the inverse convolution layer to be used to recover the size of the feature map to the input image size dimension. In DeepLab [35], the dilation convolution is used to increase the perceptual field size in the network, avoiding downsampling with the generation of multiscale features. Wasp combines four different dilation convolutions with different dilation rates in parallel with pooling to recover the feature map at the resolution of the original image. In Omnipose [36], the dilation convolution is combined with a waterfall structure. wasp architecture creates waterfall streams by using four different dilation convolutions with different dilation rates. And wasp combines all branches of the stream with the average pooling of the original input to achieve a multiscale representation. Dilated convolution increases the perceptual field of the network and generates multiscale features. We use a densely connected architecture in SDHRNet and fuse low-order features to exploit this feature in a more efficient way.

2.3 Lightweight Network

Achieving a balance between the number of parameters and accuracy in deep neural network architectures has been a critical research area. The current solutions revolve around lightweight networks or model compression techniques, such as knowledge distillation and pruning. Lightweight networks use lightweight ideas in model design to improve convolutional neural networks. Techniques such as lightweight convolutions, global pooling, and 1×1 convolutions have been used to reduce the computation of models. MobileNetV1 [27] constructs lightweight network models by replacing regular convolution with Depthwise Separable Convolution (DSC) [27]. ShuffleNet [26] addresses the drawbacks caused by group convolution by shuffling different channels and significantly reduces model computation while maintaining accuracy. The development of lightweight networks is of significant importance in convolutional neural network research. They can be used as backbone networks in various fields and can aid in the development of downstream tasks.

3 Method

The main aim of this study is to predict 2D human key points in input images and explain the overall correlation between these keypoint predictions. We propose SDHRNet, a novel network architecture depicted in Fig. 2 to achieve this. The architecture consists of lightweight feature encoders, a semantic enhancement module SEM and a spatial pyramid pool structure DASP. Initially, low-level features are extracted from the input image, which is then fed into a multi-resolution parallel backbone network consisting of lightweight feature encoders and a semantic enhancement module. The subsequently generated feature maps are processed together with the low-level features by the DASP module, which implements a spatial pyramid pool structure. This module generates n heatmaps, where n is the number of key points. Our network maintains high resolution and reduces the number of parameters and computational costs.

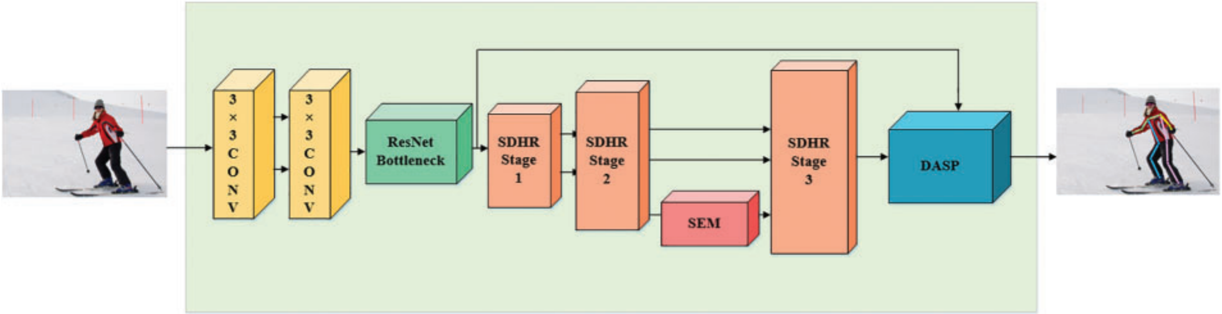

Figure 2: Illustration of SDHRNet architecture. The input images are initially fed into two 3 × 3 convolutional and bottleneck layers to extract low-order features. They are then fed into a modified backbone network based on the HRNet multi-resolution network. The backbone network consists of the SDHRStage and the semantic enhancement module SEM, which is implemented as shown in Fig. 3. The feature maps output from the backbone network is fed into the dense atrous spatial pyramid pooling module DASP along with the low-order features to finally generate n heatmaps

The SDHRStage consists of the lightweight feature encoder SCR, which is implemented as shown in Fig. 3. From SDHRStage1 to SDHRStage3, compared with the previous stage, each stage generates an additional low-resolution subnet with resolutions of 1/2, 1/4, and 1/8 of the low-order features, respectively, and the multi-resolution subnets are connected in parallel, and the information of different resolutions are fused. When the feature map is input from SDHRStage2 to SDHRStage3, the deepest feature map is enhanced by the semantic enhancement module to enhance the global feature extraction capability. The final SDHRStage3 outputs the fused feature maps and passes them to the DASP module for processing, which generates n heatmaps corresponding to the number of key points in the dataset.

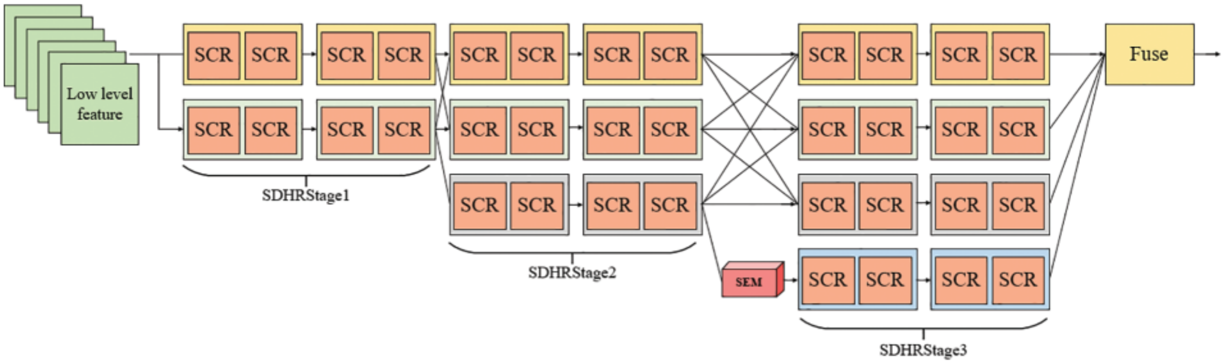

Figure 3: Illustration of multi-resolution backbone network. The network has three SDHRstages, which consist of the lightweight feature encoder SCR, each of which generates an additional low-resolution subnet with resolutions of 1/2, 1/4, and 1/8 of the low-order feature resolution, respectively

3.1 Lightweight Feature Encoder SCR

Convolutional neural networks usually consist of a large number of convolutions, which can impose a large computational cost. Although recent works such as ShuffleNet [26] and MobileNet [27] introduce shuffle operations or deep convolution to build efficient CNNs, the remaining 1 × 1 convolutional layers still make the network with considerable memory and FLOPs. To reduce the number of parameters and computational complexity of the network, we propose a lightweight feature encoder SCR (Semi-channel depth convolution—channel redistribution residual structure), as shown in Fig. 4, where the feature encoder consists of two cascaded SC (Semi-channel depthwise convolution—channel redistribution) modules to form the residual structure.

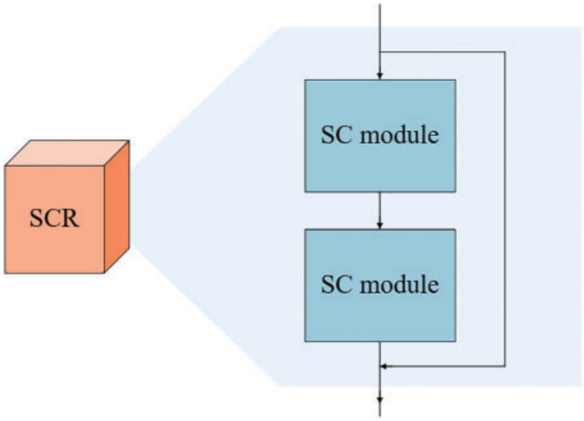

Figure 4: Lightweight feature encoder structure. This structure is a residual structure composed of two SC modules

(1) Structure: The SC module is shown in Fig. 5 and consists of a half-channel depthwise convolution unit and a channel weight redistribution unit.

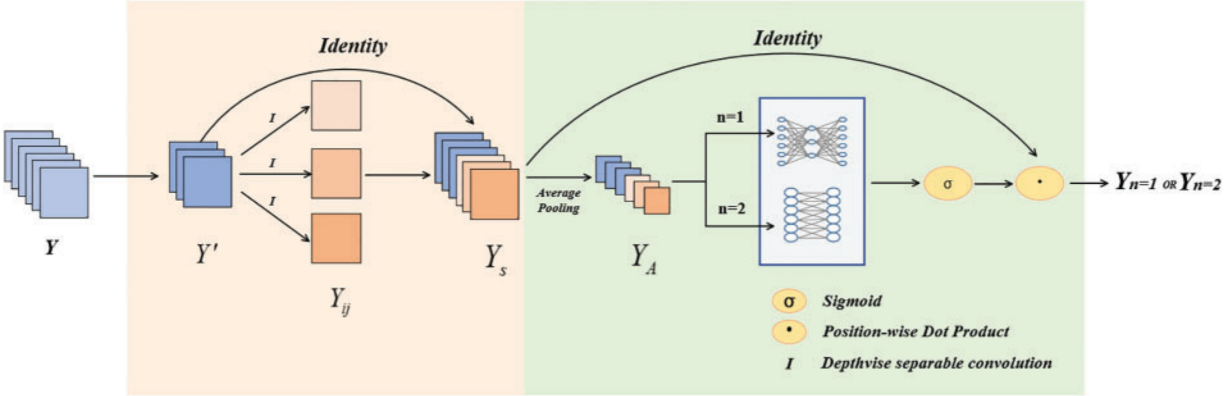

Figure 5: SC module structure. The pink part is the half-channel depthwise convolution unit and the green part is the channel weight redistribution unit. The two parts are connected in series to obtain the SC module

In practice, given the input data $x \in R_{c\times h\times w}$, where h and w are the height and width of the input feature map, respectively, and c is the number of channels. Then the regular convolution operation is $Y = X \cdot f$, where Y is the output feature, f is the convolution kernel, and the convolution kernel size is k. In the SC module, the input feature map is first subjected to regular convolution to obtain the intermediate feature map Y' with the number of channels halved.

$$Y' = X \cdot f' \tag{1}$$

where f' is the convolution kernel, $f' \in R^{c\cdot k\cdot k\cdot \frac{c}{2}}$, and $k \cdot k$ is the convolution kernel size. Then Y' obtains the half-channel feature map Y_{ij} by depthwise convolution I,

$$Y_{ij} = I_{ij}\left(Y'\right), \forall \text{ i}, j = 1, 2, \ldots, c/2 \tag{2}$$

where I_{ij} means the i-th intermediate feature map obtains the j-th half-channel feature map by depthwise convolution I.

Finally, the intermediate feature map is superimposed with the half-channel feature map to obtain the output feature map Y_s.

$$Y_s = Y' + Y_{ij} \tag{3}$$

The above structure is a half-channel depth convolution unit. The feature map obtained by the half-channel depth convolution unit has the same number of channels as the input feature map.

Since depthwise convolution does not fully consider the relationship between individual channels, channel weight reallocation is needed to boost the channel weights of feature maps that are useful for the current task and suppress the feature channels that are not useful for the current task. The feature maps are input to the channel weight redistribution unit, and firstly, the average pooling is performed to obtain a vector Y_A of size $1 \times 1 \times c$,

$$Y_A = F_A\left(Y_s\right) \tag{4}$$

where F_A is the average pooling operation.

When the module is the first SC unit in SCR, then n = 1, the global learning of the channel feature map is performed using two fully connected layers, where the first fully connected layer is used for dimensionality reduction and the second fully connected layer is used to recover the original scale. When the module is the second SC unit in SCR, then n = 2, and local channel features are extracted using adaptive one-dimensional convolution. The size of the convolution kernel k_c2 is proportional to the number of channels. The adaptive function is as follows:

$$k_{c2} = \left|\frac{log_2\left(c\right)}{\gamma} + \frac{b}{\gamma}\right|, \gamma = 2, b = 1 \tag{5}$$

The sigmoid operation is performed on the output vector of $1 \times 1 \times \text{c}$. Then the output is multiplied with the half-channel depth convolutional feature map Y_s and the channel weights are reassigned to obtain the feature map $Y_{n=1}$ or $Y_{n=2}$.

$$Y_{n=1} = \sigma\left(W_2 RELU\left(W_1 Y_A\right)\right) \cdot Y_s \tag{6}$$

$$Y_{n=2} = \sigma\left(D_k\left(Y_A\right)\right) \cdot Y_s \tag{7}$$

Two SC units form the residual structure to obtain the SCR module.

(2) Parameter calculation: The depthwise convolution runs on each channel with a much smaller number of parameters than the normal convolution. The final number of parameters for a half-channel depth convolution unit is formulated as follows:

$$P_s = c \times \frac{c}{2} \times k \times k + k' \times k' \times \frac{c}{2} \tag{8}$$

The operational complexity is formulated as follows:

$$G_s = H \times W \times \left(c \times \frac{c}{2} \times k \times k + k' \times k' \times \frac{c}{2}\right) \tag{9}$$

where k' is the size of the convolution kernel for the depthwise convolution.

The number of parameters and the computation of the channel weight reassignment unit are as follows. The number of parameters when n equals 1

$$P_{c_1} = 2 \times c \times c_{mid} \tag{10}$$

The operational complexity is

$$G_{c_1} = 2 \times c \times c_{mid} \tag{11}$$

For $n = 2$, the number of parameters and the complexity of the operation are negligible.

The final number of parameters of the whole lightweight feature encoder is:

$$P = P_s + P_{c_1} + P_s + P_{C_2} \approx 2P_s + P_{c_1} = 2 \times \left(c \times \frac{c}{2} \times k \times k + k' \times k' \times \frac{c}{2}\right) + 2 \times c \times c \times \frac{1}{4}$$

$$= \left(k^2 + \frac{1}{2}\right) c^2 + k'^2 \times c \tag{12}$$

The number of parameters for the basic module in HRNet is

$$P_0 = c \times c \times k^2 \times 2 \tag{13}$$

In the specific implementation, the selected convolutional kernel size k and K′ are both 3. The ratio of the number of parameters of the SCR to the BASIC module in HRNet [23] is

$$r = \frac{\left(k^2 + \frac{1}{2}\right) c^2 + k'^2 \times c}{c \times c \times k^2 \times 2} = 0.52 + 0.25\frac{1}{c} \approx 0.52 \tag{14}$$

It can be seen that our proposed lightweight feature encoder can effectively reduce the number of parameters of the network.

3.2 Semantic Enhancement Module

Convolutional networks only focus on the local receptive field, which has the limitation of only using local information and cannot account for global semantic information, resulting in bias [37]. This problem is generally alleviated by using larger convolutional filters or stacking more convolutional and pooling layers, which tend to increase the complexity of the network. To solve this problem, we propose a semantic enhancement module to capture the information between pixels at long distances and finally achieve a global perceptual field for each pixel, as shown in Fig. 6. The information correlation between pixels of different space and channels can also be obtained by different operations.

Channel semantic enhancement benefits keypoint identification by utilizing inter-level representation, whereas spatial semantic enhancement benefits keypoint identification by optimizing spatial information. We concatenate channel semantic information with spatial semantic information to enhance distance dependence, achieve global connectedness, and adaptively correct channel features.

Define the feature map with channel number C as X, the global average pooling as $GAP(\cdot)$, the hard sigmoid as $\sigma(\cdot)$, the one-dimensional convolution of the convolution kernel determined by the adaptive function as F, and the element multiplication as $\odot$, then the channel semantic information is calculated as follows:

$$H_{channel} = X \odot \sigma\left(F\left(GAP\left(X\right)\right)\right) \tag{15}$$

The reshape operation reshapes the feature map of $c \times h \times w$ as $c \times n$, defined as $R(\cdot)$, matrix multiplication as $\cdot$, and element-by-element summation operation as $\oplus$, where $n = h \times w$, the number of pixels. Firstly, the feature map is subjected to ordinary convolution to generate ABD. Then matrix multiplication is performed between the transpose of A and B, and the attention map E is computed by applying the softmax layer. The matrix multiplication is performed between the transpose of E and D. The final matrix is reshaped as $C \times H \times W$ and multiplied by the scale parameter, and the element-by-element summation operation is performed on the feature X to obtain the final output $H_{spatial}$:

$$H_{spatial} = X \oplus \left(R\left(R^T\left(\sigma\left(R^T\left(A\right) \cdot B\right)\right) \cdot D\right)\right) \tag{16}$$

Finally, the feature maps in the two branches are summed to obtain the output features. Our semantic enhancement module can be formulated as follows:

$$H = H_{channel} + H_{spatial} \tag{17}$$

By manipulating the space and channels, we obtain semantic enhancement modules to obtain the information correlation of different channels with the space and improve the network performance.

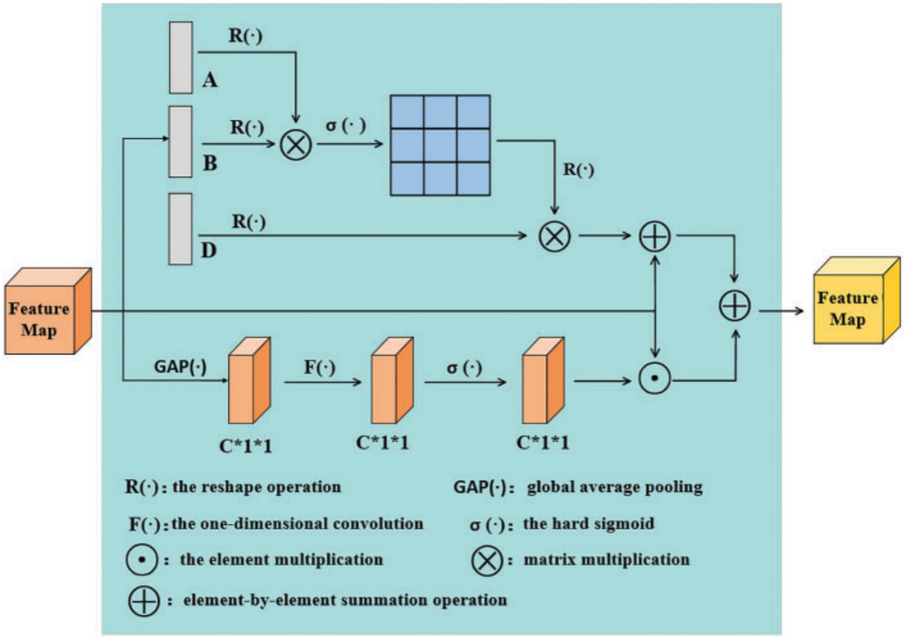
R(·)
A
R(·)
σ(·)
B
R(·)
R(·)
D
Feature Map
Feature Map
GAP(·)
F(·)
σ(·)
C*1*1
C*1*1
C*1*1
R(·): the reshape operation
GAP(·): global average pooling
F(·): the one-dimensional convolution
σ(·): the hard sigmoid
⊙: the element multiplication
⊗: matrix multiplication
⊕: element-by-element summation operation

Figure 6: Illustration of semantic enhancement module

3.3 Dense Atrous Spatial Pyramid Pooling Module (DASP Module)

We propose a dense atrous spatial pyramid pooling module DASP. As shown in Fig. 7, the DASP module densely connects the dilated convolution and fuses it with low-order features to generate an effective multiscale representation module that acquires more scale information, compensates for the reduced resolution and information loss due to downsampling in lightweight networks, which effectively improve the performance.

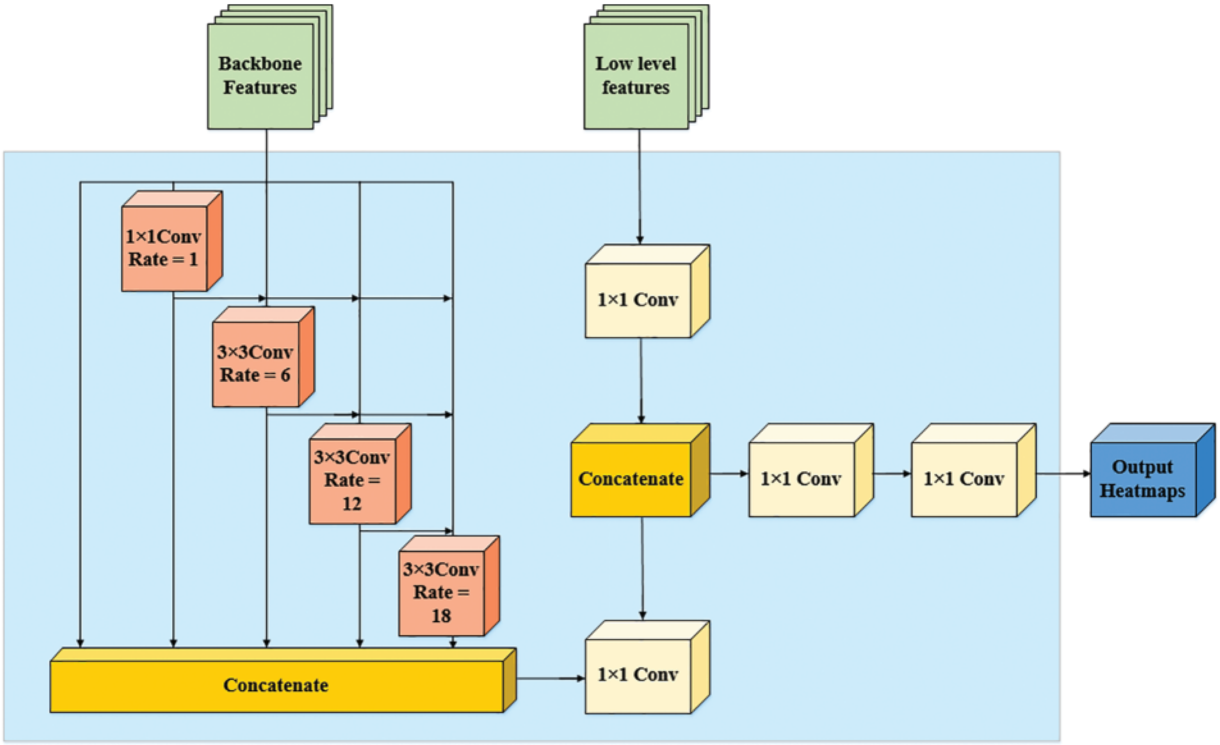

Figure 7: Illustration of DASP architecture

The DASP module relies on dilation convolution to expand the receptive field without sacrificing the spatial resolution of the features and uses a reasonable dilation rate (d < 24) to combine dilation convolution features with different dilation rates. By densely connecting the outputs of each dilation convolution, a larger range of dilation rates is obtained, and the later neurons acquire larger and larger receptive fields while avoiding the convolutional degradation caused by convolution with too large dilation rates. By combining a series of dilated convolutions, the neurons on the feature maps encode semantic information at multiple scales, and different intermediate feature maps encode information from different scales. By fusing low-order features, a more accurate and fine-grained response can be obtained. The final output features of DASP not only cover a large range of semantic information, but also extract features in a very dense manner. The output of the DASP module is described as follows.

$$y_l = H_{k,d_l}\left([y_{l-1}, y_l - 2, \ldots, y_0]\right) \tag{18}$$

$$f_{DASP} = K_1 \circledast K_1 \circledast (K_1 \circledast f_{LLF} + K_1 \circledast y_l) \tag{19}$$

where y_l is the output result of the dense connection, H_k is the expansion convolution operation on the feature map, k is the convolution kernel size, d_l is the expansion rate of the l layer. $[\cdot]$ is a cascade operation. $[y_{l-1}, y_l - 2, \ldots, y_0]$ is that the feature map of this layer is obtained by cascading the output of the upper layer. f_{DASP} is the output result after fusing the low-order features, f_{LLF} refers to the low-order features, $\circledast$ is the standard convolution operation on the feature map, K_1 is the normal convolution with a convolution kernel of 1×1, $+$ is the concat operation.

The DASP module maintains the spatial resolution of the network and combines dilated convolutional features with different dilation rates. It makes the feature resolution on the scale axis denser and obtains a larger receptive field. According to the paper [38], for a dilated convolutional layer with dilation rate d and convolution kernel size k, the equivalent receptive field size is: $R = (d-1)(K-1)+K$, the receptive field after superimposing the two convolutional layers is $K = K_1+K_2$. Therefore, when the expansion rate is selected as [1,6,12,18], the receptive field of y_l in the DASP module is $13+25+37=75$.

We fuse the dense output with low-level features and finally directly output a heatmap corresponding to the number of body keypoints.

4 Experiments

4.1 Datasets and Evaluation Metric

Datasets. The MPII [28] and COCO [29] datasets are used for training and testing.

MPII dataset consists of images extracted from YouTube videos, with roughly 25K images, of which 40K human instances are labeled with 16 human key points. About 28,000 human instances are used as training samples, and about 2,900 human instances are used as calibration samples. For the MPII dataset [28], this paper uses the PCKh (Head-normalized Probability of Correct Keypoint) metric as the evaluation criterion. PCKh is defined as the proportion of correctly estimated keypoints, that is, the proportion of the normalized distance between the computationally detected keypoints and their corresponding true values is less than the set threshold. The set threshold is denoted by αl, where α is a constant and l denotes 60% of the diagonal length of the bounding box of the reference head. PCKh@0.5 (i.e., $\alpha = 0.5$) was used as the evaluation criterion.

COCO dataset contains over 200,000 images and 250,000 person instances labeled with 17 keypoints. Our model is trained on the train2017 set (includes 57 K images and 150 K person instances) and validated on the val2017 set (includes 5 K images). For the COCO2017 [29] dataset, the OKS (Object Keypoint Similarity) metric was experimentally used as the evaluation criterion. The OKS formula is shown below.

$$OKS = \frac{\Sigma_i exp\left(-\mathrm{d}_i^2/2s^2k_i^2\right)\delta\left(v\mathrm{i} > 0\right)}{\Sigma_\mathrm{i}\delta\left(v_i > 0\right)} \tag{20}$$

where d is the Euclidean distance between the predicted key point and the true key point, i is the number of the key point, s is the target ratio, v_i is the sign bit of the true key point, and k_i is a constant used by different key points to control the decay. The value of OKS is taken between [0, 1], and the closer the prediction is to the original value, the closer it tends to 1, and vice versa. The OKS is selected in the experiment as follows: AP^{50}, AP^{75} are the prediction accuracy when the OKS is taken as 0.5, 0.75 respectively, mAP is the prediction accuracy when the OKS is taken as 0.5, 0.55, ..., 0.90, 0.95. AP^{M}, AP^{L} are the prediction accuracies for identifying key points for medium-sized and large-sized characters, respectively.

4.2 Experimental Settings

All experiments were performed using PyTorch 1.10 running on Ubuntu 16.04. We use the stepwise method to calculate the learning rate, which starts at 10^{-3} and decreases by one order of magnitude in two steps for each of the 170 and 200 periods. Since the sizes of the original images in COCO dataset [29] and MPII [28] dataset are different, the images need to be pre-processed first. In this paper, the experiments adopt to crop the original images in COCO dataset to 256 × 192 images uniformly and the original images in MPII dataset to 256 × 256 size uniformly. And the data enhancement is achieved by using randomly scaled images and randomly flipped images.

4.3 Comparison with Existing Methods

Results on MPII. The experimental results on the MPII dataset [28] are shown in Table 1. The models in Table 1 are compared in terms of performance without loading the pre-trained models since the SDHRNet models are not pre-trained on the Image Net dataset. From Table 1, it can be seen that (i) compared to HRNet [23] on the MPII dataset [28], SDHRNet uses 41% fewer model parameters, has 36% less computational complexity, and achieves the highest detection accuracy for the shoulder, elbow, hip, knee, and ankle. (ii) compared to Hourglass, Simple Baselines and DPIT-B, the accuracy of SDHRNet is increased while the number of parameters and operational complexity is decreased, respectively. (iii) compared with Lite-HRNet [25] and Simple Baselines [20] with MobileNetV2 [27] and ShuffleNetV2 [26] as the backbone network, our network improves AP by 4.5, 7.1, 4.0, 2.9 points, while the number of parameters and ours GFLOPs is slightly larger. The results show that the SDHRNet model proposed in this paper effectively reduces the number of network model parameters and operational complexity, while improving detection accuracy.

Table 1: Comparison on MPII val set

Method	Backbone	#Params	GFLOPs	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCKh
CMU Pose [38]	-	-	-	92.4	90.4	80.9	70.8	79.5	73.1	66.5	79.1
SPM [37]	-	-	-	92.0	88.5	78.6	69.4	77.7	73.8	63.9	77.7
RMPE [39]	-	-	-	88.4	86.5	78.6	70.4	74.4	73.0	65.8	76.7
Hourglass [22]	Hourglass	25.1 M	19.1	96.5	95.3	88.4	82.5	87.1	83.5	78.3	87.5
Simple baselines [20]	Simple baselines	68.6 M	20.9	96.7	95.4	88.6	82.9	87.5	83.8	79.0	87.9
HRNet [23]	HRNet	28.5 M	9.5	97.0	95.5	90.0	**85.2**	88.1	85.1	81.0	89.3
DPIT-B [24]	-	21.6 M	-	**97.1**	95.7	90.0	84.6	89.4	85.9	80.7	89.6
Simple baselines [20]	MobileNetV2 [27]	9.6 M	2.12	95.3	93.5	85.8	78.5	85.9	79.3	74.4	85.4
Simple baselines [20]	ShuffleNetV2 [26]	7.6 M	1.83	94.6	92.4	83.0	75.6	82.8	75.9	69.2	82.8
Lite-HRNet [25]	Lite-HRNet-18	1.1 M	0.27	96.1	93.7	85.5	79.2	87.0	80.0	75.1	85.9
Lite-HRNet [25]	Lite-HRNet-30	1.8 M	0.42	96.3	94.7	87.0	80.6	87.1	82.0	77.0	87.0
SDHRNet	SDHRNet	16.8	6.0	97.0	**96.0**	**90.3**	85.7	**89.1**	**85.9**	**81.9**	**89.9**

Results on COCO. We performed training and testing on the COCO dataset [29], which is more challenging compared to MPII [28] due to the large number of different types of images in the COCO dataset [29], including images without human instances. Both SDHRNet and SOTA methods do not load pre-trained models and perform experiments on the coco validation set.

From Table 2, it can be seen that (i) compared with SOTA methods for the validation set of the COCO dataset [29], SDHRNet achieves better performance with less number of parameters and complexity of operations. The modification of the HRNet backbone combined with the DASP module resulted in a significant reduction in the number of parameters and operations, with a significant 41% reduction in the number of parameters and 36.6% reduction in the number of operations compared to the original HRNet [23], with accuracy remaining essentially unchanged; and the AP^{50} obtained for SDHRNet showed a better performance than the SOTA. (ii) compared with Hourglass, Cascaded Pyramid Network (CPN) [40], CPN+OHKM (Online Hard Keypoints Mining), and Simple Baseline, the mAP in the SDHRNet model in predicting keypoints increase by 6.3, 4.6, 3.8, and 2.8 percentage points, respectively, which has fewer parameters. (iii) compared with lightweight architectures such as Simple Baselines [20] with MobileNetV2 [27] and ShuffleNetV2 [26] as the backbone, Lite-HRNet-18 [25], Lite-HRNet-30 [25] and LitePose-XS [41], our network, with slightly larger model size and slightly higher complexity, achieves 8.6, 13.3, 8.4, 6.0 and 32.6 points gain, respectively. And all of SDHRNet's validation criteria *OKS* are better than these five models. Fig. 8 shows the comparison of FLOPs and accuracy in experiments performed on the MPII (left) and COCO (right) datasets. Example results for the validation COCO dataset [29] are shown in Fig. 9.

Table 2: Comparison on COCO2017 val set

Method	Backbone	#Params	GFLOPs	mAP/%	AP^{50}/%	AP^{75}/%	AP^{M}/%	AP^{L}/%	AR/%
Hourglass [22]	Hourglass	25.1	14.3	66.9	-	-	-	-	-
CPN [40]	ResNet-50 [21]	27.0	6.2	68.6	-	-	-	-	-
CPN+OHKM	ResNet-50 [21]	27.0	6.2	69.4	-	-	-	-	-
Simple baselines [20]	ResNet-50 [21]	34.0	8.9	70.4	88.6	78.3	67.1	77.2	76.3
HRNet [23]	HRNet [23]	28.5	7.1	73.4	89.5	80.7	70.2	80.1	78.9
Simple baselines [20]	MobileNetV2 [27]	9.6	1.59	64.6	87.4	72.3	61.1	71.2	70.7
Simple baselines [20]	ShuffleNetV2 [26]	7.6	1.37	59.9	85.4	66.3	56.5	66.2	66.4
Lite-HRNet [25]	Lite-HRNet-18	1.1	0.2	64.8	86.7	73.0	62.1	70.5	71.2
Lite-HRNet [25]	Lite-HRNet-30	1.8	0.3	67.2	88.0	75.0	64.3	73.1	73.3
LitePose-XS [41]	-	1.7	-	40.6	-	-	-	-	-
SDHRNet	SDHRNet	16.8	4.5	73.2	**89.6**	80.7	69.8	79.9	78.7

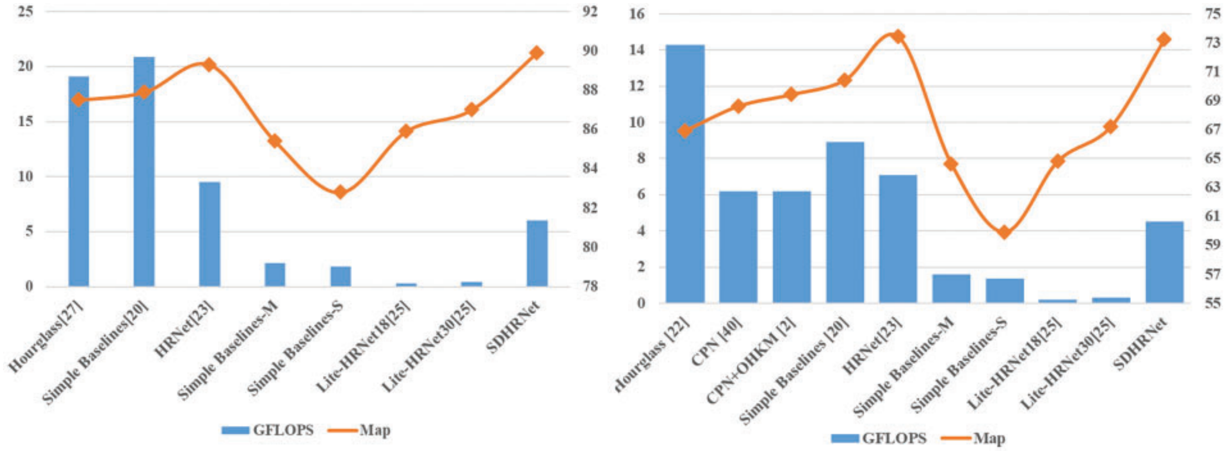

Figure 8: The comparison of FLOPs and accuracy in the experiments on MPII (left)and COCO (right) datasets

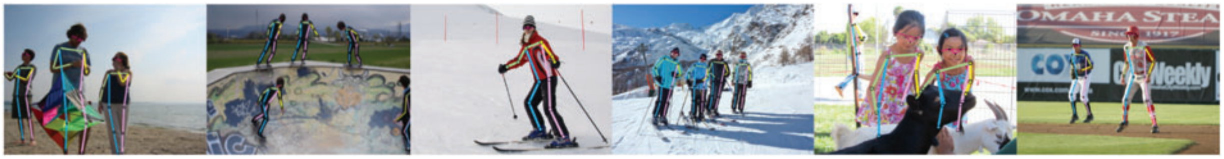

Figure 9: Prediction results on COCO val

4.4 *Ablation Study*

In the experiments on the COCO dataset, we performed a series of ablation studies to validate the feature extraction capability of the added individual modules and analyze the gains obtained by the different modules. Table 3 demonstrates the results for the inclusion of the Lightweight encoder in the HRNet backbone, DASP module and Semantic Enhancement Module. After adding SCR, the number of parameters and FLOPs in the network decreases while the mAP slightly decreases. DASP and SEM increased the mAP of the network, while the number of parameters and FLOPs slightly increased.

Table 3: Ablation experiments of each component on COCO2017 val set

Method	SCR	DASP	SEM	#Params	GFLOPs	mAP/%
HRNet				28.5	9.5	73.4
SDHRNet	√			16.7	4.4	71.5
SDHRNet	√	√		16.7	4.5	73.0
SDHRNet	√	√	√	16.8	4.5	73.2

Our method has gradually improved its performance by adding innovations, and the results are almost on par with HRNet [23]. Most importantly, the lightweight encoder greatly reduces the number of parameters and FLOPs of the network. And the DASP and semantic enhancement modules compensate for the information and accuracy loss caused by lightweight feature encoders with a small number of parameters, restoring the final model's average accuracy to HRNet [23] levels.

5 Conclusion

In this paper, we proposed SDHRNet, a lightweight multi-resolution network for human pose estimation. The multi-resolution network was built by a lightweight feature encoder, which reduced the number of parameters and FLOPs of the network. At the same time, a semantic enhancement module SEM and spatial pyramid pooling module DASP were introduced. The SEM module improved global feature extraction and network performance by combining both channel and spatial dimensions. The DASP module tightly connected the dilation convolution and fuses it with the low-level features to obtain more scale information and compensate for the reduced resolution and information loss due to downsampling in the lightweight network. Experiments on the MPII dataset and COCO dataset validated that the network proposed in this paper achieved a competitive performance in the human pose estimation task with a balance of accuracy and the number of parameters. In future, reducing the number and accuracy of network parameters while improving the localization accuracy of key points is an important follow-up research goal.

Acknowledgement: This work was supported by the National Natural Science Foundation of China. The authors would like to thank the editors and the reviewers for their useful feedback that improved this paper.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant Number 62076246).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Pengxin Li; data collection: Pengxin Li, Wenjing Zhang, Yinuo Liu, Chenyue Xu; analysis and interpretation of results: Pengxin Li, Rong Wang; draft manuscript preparation: Pengxin Li, Wenjing Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chen, Y., Tian, Y., He, M. (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding, 192,* 1077–3142.
2. Lin, J., Li, S., Qin, H., Wang, H., Cui, N. et al. (2023). Overview of 3D human pose estimation. *Computer Modeling in Engineering & Sciences, 134(3),* 1621–1651. https://doi.org/10.32604/cmes.2022.020857
3. Ishwarya, K., Alice Nithya, A. (2023). Squirrel search optimization with deep convolutional neural network for human pose estimation. *Computers, Materials & Continua, 74(3),* 6081–6099. https://doi.org/10.32604/cmc.2023.034654
4. Kumarapu, L., Mukherjee, P. (2021). AnimePose: Multi-person 3D pose estimation and animation. *Pattern Recognition Letters, 147,* 16–24.
5. Bridgeman, L., Volino, M., Guillemaut, J. Y., Hilton, A. (2019). Multi-person 3D pose estimation and tracking in sports. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA.
6. Yoo, H., Chung, K. (2022). Classification of multi-frame human motion using CNN-based skeleton extraction. *Intelligent Automation & Soft Computing, 34(1),* 1–13. https://doi.org/10.32604/iasc.2022.024890
7. Arif, A., Yasin Ghadi, Y., Alarfaj, M., Jalal, A., Kamal, S. et al. (2022). Human pose estimation and object interaction for sports behaviour. *Computers, Materials & Continua, 72(1),* 1–18. https://doi.org/10.32604/cmc.2022.023553
8. Li, T., Chen, J., Hu, C., Ma, Y., Wu, Z. et al. (2018). Automatic timed up-and-go sub-task segmentation for Parkinson's disease patients using video-based activity classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26,* 2189–2199.
9. Tompson, J. J., Jain, A., LeCun, Y., Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 27, pp. 1799–1807. Montreal, Canada.
10. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C. (2015). Efficient object localization using convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648–656. Boston, MA, USA.
11. Wei, S. E., Ramakrishna, V., Kanade, T., Sheikh, Y. (2016). Convolutional pose machines. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732. Las Vegas, NV, USA.
12. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C. (2020). Distribution-aware coordinate representation for human pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7093–7102. Seattle, WA, USA.
13. Isack, H., Haene, C., Keskin, C., Bouaziz, S., Boykov, Y. et al. (2020). Repose: Learning deep kinematic priors for fast human pose estimation. arXiv:2002.03933.
14. Li, Y., Yang, S., Zhang, S., Wang, Z., Yang, W. et al. (2021). Is 2D heatmap representation even necessary for human pose estimation? arXiv:2107.03332.
15. Toshev, A., Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660. Columbus, OH, USA.
16. Geng, Z., Sun, K., Xiao, B., Zhang, Z., Wang, J. (2021). Bottom-up human pose estimation via disentangled keypoint regression. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14676–14686.

17. Sun, X., Shang, J., Liang, S., Wei, Y. (2017). Compositional human pose regression. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2602–2611. Honolulu, HI, USA.

18. Li, J., Bian, S., Zeng, A., Wang, C., Pang, B. et al. (2021). Human pose regression with residual log-likelihood estimation. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 11025–11034. Montreal, Canada.

19. Wei, F., Sun, X., Li, H., Wang, J., Lin, S. (2020). Point-set anchors for object detection, instance segmentation and pose estimation. *Proceedings of the European Conference on Computer Vision*, pp. 527–544. Glasgow, UK.

20. Xiao, B., Wu, H., Wei, Y. (2018). Simple baselines for human pose estimation and tracking. *Proceedings of the European Conference on Computer Vision*, pp. 466–481. Munich, Germany.

21. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. Las Vegas, NV, USA.

22. Newell, A., Yang, K., Deng, J. (2016). Stacked hourglass networks for human pose estimation. *Proceedings of the European Conference on Computer Vision*, pp. 483–499. Amsterdam, The Netherlands.

23. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C. et al. (2021). Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43,* 3349–3364.

24. Zhao, S., Liu, K., Huang, Y., Bao, Q., Zeng, D. et al. (2023). DPIT: Dual-pipeline integrated transformer for human pose estimation. *Artificial Intelligence: Second CAAI International Conference*, pp. 559–576. Beijing, China.

25. Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L. et al. (2021). Lite-HRNet: A lightweight high-resolution network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10440–10450.

26. Ma, N., Zhang, X., Zheng, H. T., Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient cnn architecture design. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 116–131. Salt Lake City, UT, USA.

27. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. Salt Lake City, UT, USA.

28. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693. Columbus, OH, USA.

29. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P. et al. (2014). Microsoft COCO: Common objects in context. *Proceedings of the European Conference on Computer Vision*, pp. 740–755. Zurich, Switzerland.

30. Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T. (2012). Articulated people detection and pose estimation: Reshaping the future. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3178–3185. Providence, RI, USA.

31. Wang, S. H., Govindaraj, V. V., Górriz, J. M., Zhang, X., Zhang, Y. D. (2021). COVID-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Information Fusion, 67,* 208–229.

32. Zhang, Y. D., Yang, Z. J., Lu, H. M., Zhou, X. X., Phillips, P. et al. (2016). Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. *IEEE Access, 4,* 8375–8385.

33. Wang, S., Zhang, Y., Dong, Z., Du, S., Ji, G. et al. (2015). Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection. *International Journal of Imaging Systems and Technology, 25(2),* 153–164.

34. Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440. Boston, MA, USA.

35. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40,* 834–848.

36. Artacho, B., Savakis, A. (2021). Omnipose: A multiscale framework for multi-person pose estimation. arXiv:2103.10180.

37. Nie, X., Feng, J., Zhang, J., Yan, S. (2019). Single-stage multi-person pose machines. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6951–6960. Long Beach, CA, USA.

38. Cao, Z., Simon, T., Wei, S. E., Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299. Honolulu, HI, USA.

39. Fang, H. S., Xie, S., Tai, Y. W., Lu, C. (2017). Rmpe: Regional multi-person pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2334–2343. Honolulu, HI, USA.

40. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G. et al. (2018). Cascaded pyramid network for multi-person pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112. Salt Lake City, UT, USA.

41. Wang, Y., Li, M., Cai, H., Chen, W. M., Han, S. (2022). Lite pose: Efficient architecture design for 2D human pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13126–13136. New Orleans, USA.