

Fusion of Convolutional Self-Attention and Cross-Dimensional Feature Transformation for Human Posture Estimation

Anzhan Liu[✉], Yilu Ding[✉], Xiangyang Lu

Abstract: Human posture estimation is a prominent research topic in the fields of human-computer interaction, motion recognition, and other intelligent applications. However, achieving high accuracy in key point localization, which is crucial for intelligent applications, contradicts the low detection accuracy of human posture detection models in practical scenarios. To address this issue, a human pose estimation network called AT-HRNet has been proposed, which combines convolutional self-attention and cross-dimensional feature transformation. AT-HRNet captures significant feature information from various regions in an adaptive manner, aggregating them through convolutional operations within the local receptive domain. The residual structures TripNeck and TripBlock of the high-resolution network are designed to further refine the key point locations, where the attention weight is adjusted by a cross-dimensional interaction to obtain more features. To validate the effectiveness of this network, AT-HRNet was evaluated using the COCO2017 dataset. The results show that AT-HRNet outperforms HRNet by improving 3.2% in mAP, 4.0% in AP⁷⁵, and 3.9% in AP^M. This suggests that AT-HRNet can offer more beneficial solutions for human posture estimation.

Keywords: human posture estimation; adaptive fusion method; cross-dimensional interaction; attention module; high-resolution network

1 Introduction

In recent years, human pose estimation (HPE) has gained significant attention as a challenging research problem in the field of computer vision. HPE aims to predict a person's pose by detecting key points in still images or videos. Currently, HPE has found applications in various

fields, including healthcare [1–2], motion analysis [3], driver posture detection [4–5], and virtual fitting [6].

Traditional methods for human pose estimation typically utilize image features based on graphical structures [7] and deformable part models [8] to represent various key points as part detectors containing spatial constraint relationships, such as HOG [9], SIFT [10], etc. Subsequently, artificially designed feature templates are employed to detect these key points. For instance, Yang et al. [11] employed flexible mixtures-of-parts to express the spatial constraint relationships of human pose. Pishchulin et al. [12] proposed a tree-structured spatial model conditioned on images to optimize pose estimation algorithms. Traditional pose estimation methods are characterized by high time efficiency, but

Manuscript received Jan. 31, 2024; revised Apr. 30, 2024; accepted May 30, 2024. The associate editor coordinating the review of this manuscript was Dr. Xuejing Kang. This work was supported by the National Natural Science Foundation of China (No. 61975015) and the Research and Innovation Project for Graduate Students at Zhongyuan University of Technology (No. YKY2024ZK14).

Anzhan Liu and Yilu Ding are with School of Computer College, Zhongyuan University of Technology, Zhengzhou 451191, China.

Xiangyang Lu is with School of Electronic and Information College, Zhongyuan University of Technology, Zhengzhou 451191, China.

✉ Corresponding author. Email: liuanzhan@zut.edu.cn, yilu_ding@126.com

DOI: 10.15918/j.jbit1004-0579.2024.018

limited by data availability and other factors such as varying lighting conditions, complex backgrounds, and occlusions. Furthermore, these methods typically employ a single model structure, resulting in poor scalability and relatively low accuracy. With the development of deep learning, the use of convolutional neural networks (CNN) [13] for human posture estimation has emerged as a new research focus. CNN gradually extracts high-level features from local image features through layer-by-layer convolutional and pooling operations, further obtaining global feature information [14–15]. By extracting features advantageous for keypoints and context-based information from various scales of receptive fields, it facilitates a more precise calculation of the positions of target keypoints. Compared to traditional methods, CNN-based approaches overcome the limitations of a single model and enable better utilization of image information and keypoint attributes. In CNN-based methods, the size of the convolutional receptive field plays a crucial role in pose estimation tasks. A small receptive field may result in a lack of correlation among joints, while a large receptive field can introduce unnecessary information, potentially reducing attention to the main features.

The attention mechanism draws inspiration from the human behavior of actively focusing on pertinent information while disregarding irrelevant details during observations. It has found widespread application in various domains, including image processing, natural language processing, and more. This mechanism has played a significant role in the field of computer vision. Within the convolutional neural network model, the attention mechanism enables a higher focus on essential features, thereby enhancing the model’s accuracy. Scholars have researched various aspects related to the attention mechanism, such as determining what the attention mechanism should prioritize and how to extract key feature information from it.

In 2021, Misra et al. [16] proposed the Triple

Attention mechanism. This mechanism calculates attention weight values by capturing cross-dimensional interactions in a three-branch structure without any dimensionality reduction operation. As a result, it eliminates the indirect correlation between weights and channels, leading to the capture of richer feature information. In 2022, Pan et al. [17] introduced a combined convolutional and self-attention mechanism called Acmix. Their work demonstrated the potential relationship between convolution and self-attention and revealed that these operations involve similar computations. The convolution operation extracts features by weighting and summing features within a local receptive field. The parameters of the convolutional kernel are shared across the entire feature map, enabling the convolution operation to effectively utilize local feature information from images and generate feature information with inductive bias. The self-attention mechanism enhances the model’s perceptual ability and computational efficiency by capturing global dependencies, dynamically calculating attention weights, and enabling highly parallelized computation. Acmix effectively combines the advantages of convolution and self-attention mechanisms while reducing the computational overhead compared to the pure convolution or self-attention mechanism.

According to the above analysis, this paper proposes a human posture estimation network named AT-HRNet that fuses convolutional self-attention and cross-dimensional feature transformations, with HRNet being a baseline model, to improve the accuracy of keypoint detection. To capture feature information from different domains, AT-HRNet introduces AcBlock, which is a multi-resolution fusion module of convolution and self-attention mechanism. To address the interdependence between channel attention and spatial attention, the TripNeck and TripBlock modules are designed to improve the residual structure in the HRNet model. The main work of this paper is as follows:

1) A human posture estimation network named AT-HRNet is proposed, which fuses convolutional self-attention and cross-dimensional feature transformations.

2) A multi-resolution fusion module AcBlock combining convolution and self-attention is designed, which can obtain the advantages of both convolution and self-attention mechanisms based on only a small computational overhead, capturing both local and global semantic information of the high-resolution network and improving the performance of the model.

3) The TripBlock and TripNeck have been designed, combining channel attention and spatial attention. During the feature extraction phase of the network architecture, cross-dimensional interaction is applied to channel and spatial attention, thereby enhancing the capability of the model for feature extraction.

2 Related Work

2.1 Human Posture Estimation

CNN methods for human pose estimation can be broadly classified into two categories: bottom-up and top-down methods. Bottom-up methods involve detecting keypoint heatmaps for all human bodies in the image, followed by summarizing and categorizing these key points for pose estimation on a per-human-instance basis. On the other hand, top-down methods start by detecting each human instance in the image and then perform keypoint detection and pose estimation for each individual. Presently, bottom-up methods offer faster detection speed but relatively lower accuracy, while top-down methods provide higher accuracy at the expense of slower detection speed. Current research has focused on addressing these aspects in both types of methods.

DeepPose [18] was introduced as the first top-down human pose estimation method based on convolutional neural networks. DeepPose maps the pose estimation problem to a key point

direct regression problem by predicting the probability of each key point at different locations, forming a probabilistic heatmap. The actual location of the key point is estimated by finding the maximum value in the heatmap. Although DeepPose achieved good results, it faces challenges in directly regressing 2D coordinate points.

Wei et al. proposed the convolution pose machine (CPM) [19] method, which utilizes a sequential convolutional model to learn image feature representations and incorporates the ability to cascade predictive representations with spatial context information. CPM employs cascaded key point corrections to improve accuracy. CPM extracts more useful information by continuously expanding the receptive field, where the intermediate supervision is employed to enhance keypoint accuracy.

Newell et al. [20] introduced stacked hourglass networks (SHN), which resembles a stack of hourglasses. SHN is based on DeepPose and inherits the idea of multi-resolution features. It performs iterative coding and decoding operations on feature maps to better fuse global and local features. However, SHN suffers from significant information loss during the constant sampling up and down, which affects accuracy.

Carnegie Mellon University proposed OpenPose [21], a real-time human posture estimation system that utilizes a bottom-up approach. OpenPose's model network structure consists of two branches: the partial confidence map and the partial relation field. The partial confidence map predicts the heat map for each key point of the human body. The partial relation field predicts the correlation between key points by generating a unit vector for each key point. This correlation associates the key points belonging to the same person, thereby improving the efficiency of keypoint detection. On the other hand, SimpleBaseline [22] used inverse convolution to replace the upsampling operation. This approach reduces the loss of information and restores the feature map

more accurately. In 2019, Sun et al. [23] proposed the high-resolution network (HRNet).

2.2 HRNet

HRNet introduces a novel network architecture that maintains high-resolution features throughout the entire process. Unlike previous models, most of which were to first reduce the resolution until the final stage through up-sampling and then restore the high resolution to ensure that the final output of the network is a high-quality feature map, this will lead to the loss of many key features. HRNet divides the network into multiple subnets, with the high-resolution network branch as the key part, and the resolution is always consistent with the input feature map. The low-resolution network branch is added sequentially through the down-sampling method to realize the structure of parallel subnets. In the process of extracting the features of HRNet, the number of channels of each additional subnet branch is two times of the previous branch, the resolution is 1/2 of the previous branch, and the parallel branches are fused with multi-scale information by repeated stacked switching units, and the whole process always maintains the high-resolution features of the image, which ensures that the model can get a large amount of useful information during the training process, to achieve more accurate results. In recent years, there have been some relevant studies on HRNet.

Cheng et al. [24] proposed a network called HigherHRNet, based on HRNet, with stronger scale-awareness capability. HigherHRNet utilizes deconvolution for upsampling and integrates feature information from different scales effectively through a multi-scale fusion strategy, enhancing representations of human poses at various detail levels and thereby improving accuracy.

Yu et al. [25] proposed a lightweight high-resolution network, Lite-HRNet. Lite-HRNet incorporates conditional channel weights, allowing for information exchange between channels and resolutions by learning weights, striking a

balance between model complexity and computational efficiency.

Zhang et al. [26] proposed a multi-resolution human pose estimation network named GCT-nonlocal net (GNNet), combining channel attention and spatial attention mechanisms. GNNet introduces a representation fusion method based on attention mechanisms, enabling the network to extract more useful feature information to improve fusion units, thereby enhancing accuracy.

Luo et al. [27] proposed a high-resolution human pose estimation network named ENNet, which integrates dual attention mechanisms. Through the incorporation of feature information from various resolutions via the dual attention mechanism, the model's accuracy is enhanced.

2.3 Attention Mechanism

In recent years, attention mechanisms have been extensively studied due to their excellent memory capabilities. By assigning high weights to focus on primary features and reducing the influence of secondary features with low weights, models are better able to concentrate on the important parts of input information. In computer vision, the attention mechanism has been widely adopted to improve model performance [28].

Chu et al. [29] introduced the attention mechanism to human pose estimation. They proposed a context-based attention approach that combines the stacked hourglass model with the attention mechanism and incorporates conditional random fields (CRF) to model feature map correlations. Hu et al. [30] proposed a channel attention mechanism named the Squeeze-and-Excitation Network (SENet), which builds a model with a simple structure and is effective. SENet employs compression and excitation operations to obtain channel weights for the input feature layer, enabling the model to focus on learning the important parts effectively. In 2018, Woo et al. [31] proposed convolutional block

attention module (CBAM), which combines spatial attention and channel attention mechanisms to address their respective limitations. This approach achieves superior results compared to the spatial-after-channel or both-parallel methods. In the same year, Wang et al [32] introduced the self-attention mechanism to the image domain and proposed the classical non-local approach to provide a neural network component for capturing long-range dependencies. By embedding the non-local module into the network structure, the model’s perception ability and performance in capturing global information are enhanced. In 2019, Cao et al. [33] proposed the global context (GC) method. This method simplifies the spatial self-attention mechanism, offering a more concise model structure compared to SENet while accurately capturing long-range dependencies.

3 Methodology

3.1 AT-HRNet Network Structure

This paper proposes AT-HRNet, which is based on the HRNet and is a network model that integrates convolutional self-attention and cross-dimensional feature transformation. The network structure of the AT-HRNet is illustrated in Fig. 1, which is partitioned into four stages (Stage 1–4) and four branches (Branch 1–4). Each stage transforms the input feature map and

yields a new feature map. The number of input and output feature streams varies for each stage. For instance, Stage 1 takes an input feature data stream and produces two output feature data streams. On each branch, the output of the preceding stage corresponds to the input of the subsequent stage. The resolutions of input feature maps on different branches at different stages are delineated in Tab. 1.

Tab. 1 Input/output characteristic resolution of branches at each stage

Branch	Stage 1	Stage 2	Stage 3	Stage 4
Branch 1 input	256×192	64×48	64×48	64×48
Branch 2 input		32×24	32×24	32×24
Branch 3 input			16×12	16×12
Branch 4 input				8×6
Branch 1 output	64×48	64×48	64×48	64×48

Stage 1 is divided into three operational processes: Stem, AcBlock, and TripNeck. The Stem operation consists of two 3×3 convolutions with a stride of 2. The feature map, which has a size of 256×192 and 3 channels, is downsampled by a factor of 4, yielding a feature map of size 64×48 with 64 channels. Following the completion of the AcBlock operation, the feature information traverses through four stacked TripNeck residual modules. Subsequently, The feature information is passed from Branch 1 and Branch 2 to Stage 2. The features of Branch 2 are obtained through downsampling operation.

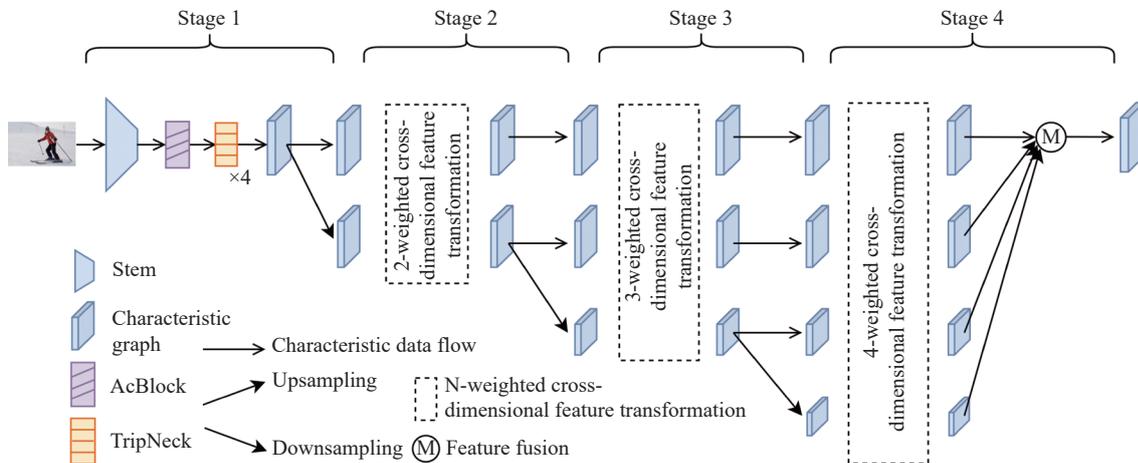


Fig. 1 AT-HRNet structure

In Stage 2, the incoming two feature data streams undergo a 2-weighted cross-dimensional feature transformation, generating new features on Branch 1 and Branch 2. The two features form the two inputs to Stage 3. The input for Stage 3 on Branch 3 is obtained from the features generated by Stage 2 on Branch 2 through downsampling. The process of Stage 3 and Stage 2 is similar.

In Stage 4, following 4-weighted cross-dimensional feature transformations, four feature data streams are produced. After upsampling in branch 2–4, the features are fused with those from branch 1 and then outputted, thereby achieving human keypoint detection.

In Stage 2–4 of the AT-HRNet, N -weighted cross-dimensional feature transformations are performed ($N = 2, 3, 4$). The difference in N primarily lies in the varying number of branches, i.e., the number of feature data streams for input and output. Here, $N = 3$ is taken as an example to provide a detailed explanation.

Fig. 2 illustrates the structure of the network for 3-weighted cross-dimensional feature transformation. The three feature data streams initially undergo TripBlock operations to acquire richer feature information during the cross-dimensional interaction process. Subsequently, multi-scale feature fusion is conducted through upsampling and downsampling, resulting in the output of three branch feature data streams. During the upsampling process, the input branch undergoes channel adjustment using 1×1 convolution with stride = 1. Batch normalization (BN) is then applied to enhance the stability and representational capability of the model.

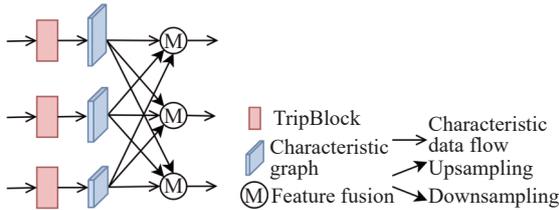


Fig. 2 3-weighted cross-dimensional feature transformation network module

Finally, nearest neighbor interpolation is utilized to upsample the feature maps, thereby enlarging the resolution. During the downsampling process, the input branch undergoes resolution adjustment using 3×3 convolution with stride = 2. BN is applied to normalize the output features, followed by ReLU activation to enhance the network’s representational power. Lastly, channel adjustment is executed through a convolutional layer to facilitate subsequent feature fusion operations.

3.2 AcBlock

Convolution and self-attention mechanisms exhibit a strong correlation. To capitalize on the shared advantages of these mechanisms, this paper designed the AcBlock structure. AcBlock can adaptively focus on different regions to acquire crucial information and aggregate data within a local receptive field using convolution. When generating the final output features, AcBlock merges the initial input features with the learned higher-level features and passes them to the next module. AcBlock significantly enhances model performance while imposing almost no additional computational overhead. The design of AcBlock is depicted in Fig. 3.

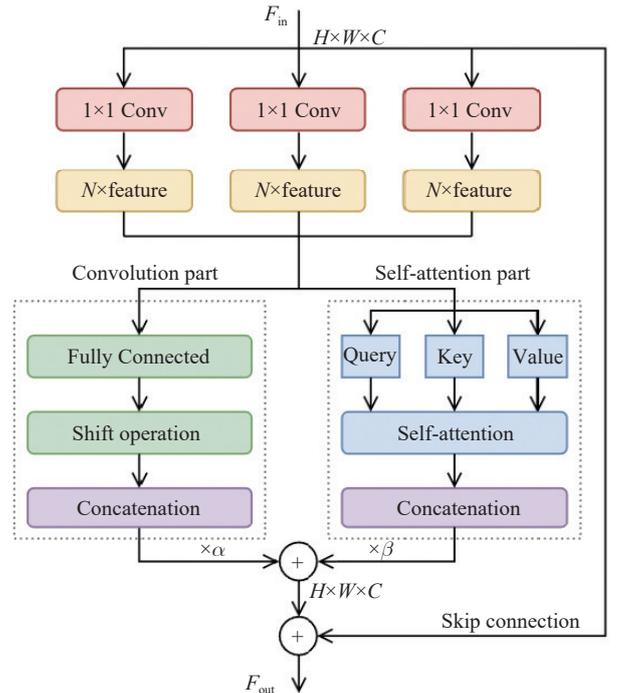


Fig. 3 AcBlock module

In AcBlock, the input feature stream is divided into four paths, with three of these paths undergoing individual 1×1 convolution operations to reconstruct them into N groups of segments, leading to the generation of N feature maps. The obtained intermediate features comprise $3 \times N$ feature maps, which are subsequently utilized as separate inputs for the convolutional part and self-attention part to undergo further processing.

Let's assume that $F \in R^{C_{in} \times H \times W}$ represents the input feature, $G \in R^{C_{out} \times H \times W}$ represents the output feature of the convolutional part, $D \in R^{C_{out} \times H \times W}$ represents the output feature of the self-attention part, and H, W represent the height and width, respectively. C_{in}, C_{out} denote the sizes of the input and output channels. The feature tensors associated with the pixels (i, j) of F, G and D are denoted as $f_{ij} \in R^{C_{in}}$, $g_{ij} \in R^{C_{out}}$ and $d_{ij} \in R^{C_{out}}$, respectively.

For the convolutional part, assuming a standard convolution kernel $K \in R^{C_{out} \times C_{in} \times k \times k}$, where k is the size of the kernel, k^2 feature maps are generated through a lightweight fully connected layer. Subsequently, shift and aggregation operations are conducted on these generated feature maps. The output expressions for the convolutional branch are provided below

$$g_{ij} = \sum_{p,q} g_{ij}^{(p,q)} \quad (1)$$

$$g_{ij}^{(p,q)} = K_{p,q} f_{i+p-\lfloor k/2 \rfloor, j+q-\lfloor k/2 \rfloor} \quad (2)$$

where $K_{p,q} \in R^{C_{out} \times C_{in}}$, $p, q \in \{0, 1, \dots, k-1\}$ denote the kernel weights relative to the kernel position (p, q) .

For Eq. (2), simplification is achieved through the Shift operation $\tilde{f} \triangleq \text{Shift}(f, \Delta x, \Delta y)$, defined as

$$\tilde{f}_{i,j} = f_{i+\Delta x, j+\Delta y}, \forall i, j \quad (3)$$

where Δx and Δy signify horizontal and vertical displacements, respectively. Consequently, Eq. (2) can be reformulated as

$$g_{ij}^{(p,q)} = \text{Shift}(K_{p,q} f_{ij}, p - \lfloor k/2 \rfloor, q - \lfloor k/2 \rfloor) \quad (4)$$

For the self-attention part, intermediate features are aggregated into N groups, with each group comprising 3 features. These three mapped features are utilized as queries, keys, and values, respectively. Attention weights are computed using the feature representations of queries, keys, and values. The output expression of the self-attention part is given by

$$d_{ij} = \parallel_{l=1}^N \left(\sum_{a,b \in \mathcal{N}_k(i,j)} A(W_q^{(l)} f_{ij}, W_k^{(l)} f_{ab}) W_v^{(l)} f_{ab} \right) \quad (5)$$

where \parallel represents the concatenation of outputs from N attention heads, while $W_q^{(l)}, W_k^{(l)}$ and $W_v^{(l)}$ are the projection matrices for queries, keys, and values, respectively. $\mathcal{N}_k(i, j)$ denotes a local region with a spatial range of K centered at (i, j) , and $A(W_q^{(l)} f_{ij}, W_k^{(l)} f_{ab})$ represents the weighting of features within $\mathcal{N}_k(i, j)$.

Next, the weighted sum of the outputs from the convolutional part and the self-attention part is computed, controlled by two learnable scalars α and β , yielding F' .

$$F' = \alpha \times g_{ij} + \beta \times d_{ij} \quad (6)$$

Finally, the initial input features F_{in} are combined with F' via skip connections and then summed to yield the output F_{out} .

$$F_{out} = F_{in} + F' \quad (7)$$

3.3 TripNeck and TripBlock

To capture the important features of spatial and channel attention, TripNeck and TripBlock are designed in this paper using the Triplet Attention mechanism, which enables the model to capture cross-dimensional interactions and thus richer features when computing the attention weights.

The processing of TripNeck goes through Triplet Attention first, followed by connecting three 2D convolutional layers, two with convolutional kernel size 1×1 and one with kernel size 3×3 . After the first two convolutional layers,

BN regularization and ReLU operations are performed, respectively. The last layer of convolution performs the BN operation, followed by residual join with the input features for residual join to perform the ReLU operation and output the feature map. The processing is shown in Fig. 4.

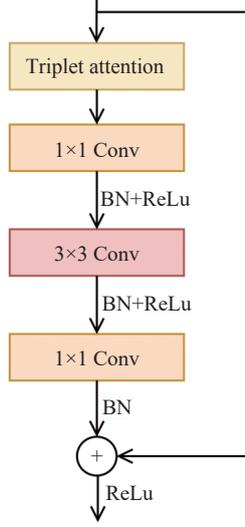


Fig. 4 TripNeck module

The processing of TripBlock is depicted in Fig. 5. First, the input features undergo the Triplet Attention operation, followed by BN regularization and the ReLU operation after passing through a 2D convolutional layer with a kernel size of 3×3 . Then, another convolution with a kernel size of 3×3 is applied, followed by BN regularization. Finally, after being connected

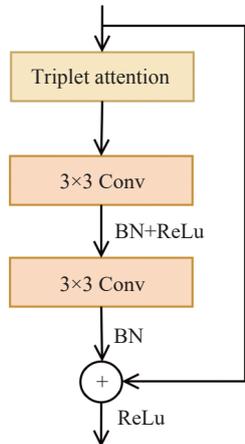


Fig. 5 TripBlock module

with the input feature residuals, a ReLU operation is performed to output the feature information.

The Triplet Attention mechanism employed in the TripNeck and TripBlock modules comprises three branches. The first and third branches are utilized to capture cross-dimensional interactions between the channel dimension C and the spatial dimensions W or H of the input tensor. The second branch specifically focuses on attention to information in the spatial dimension. The outputs of these three branches are then combined by averaging, as illustrated in Fig. 6.

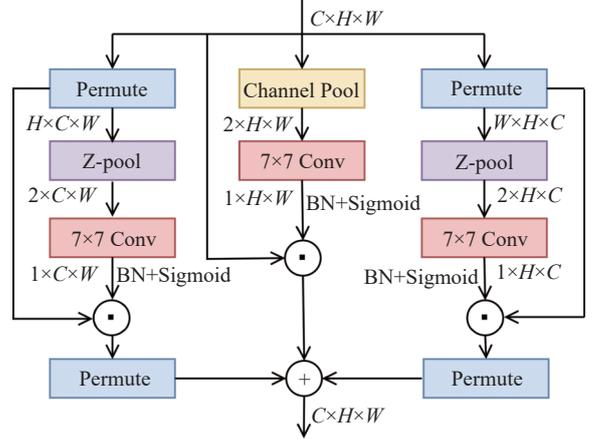


Fig. 6 Triplet attention

Given an input feature tensor $\chi \in R^{C_{in} \times H \times W}$, it serves as the input for each branch. In the first branch, interaction between the width and channel dimensions is established. χ is rotated counterclockwise by 90° along the W -axis to obtain $\widehat{\chi}_1$, with a shape of $(H \times C \times W)$. Subsequently, $\widehat{\chi}_1$ undergoes a Z-pool layer to yield $\widehat{\chi}_1^z$, simplifying its shape to $(2 \times C \times W)$. Following this, $\widehat{\chi}_1^z$ undergoes convolution with a kernel size of $k \times k$, followed by batch normalization, resulting in a shape of $(1 \times C \times W)$. Finally, $\widehat{\chi}_1^z$ is passed through a Sigmoid activation layer to generate attention weights, which are then applied to $\widehat{\chi}_1$. Afterward, $\widehat{\chi}_1$ is rotated clockwise by 90° along the W -axis to maintain the same shape as the input χ .

In the second branch, the input feature tensor χ is first processed through a Z-pool layer to obtain $\widehat{\chi}_2$ with a shape of $(2 \times H \times W)$. $\widehat{\chi}_2$ then undergoes convolution with a kernel size of $k \times k$, followed by batch normalization. Finally, attention weights of shape $(1 \times H \times W)$ are outputted after passing through a Sigmoid activation layer, and these weights are applied to the input χ .

The third branch is similar to the first branch. In the third branch, interaction between the height and channel dimensions is established. χ is rotated counterclockwise by 90° along the H -axis to obtain $\widehat{\chi}_3$ with a shape of $(W \times H \times C)$. Subsequently, $\widehat{\chi}_3$ undergoes a Z-pool layer to yield $\widehat{\chi}_3^*$ with a shape of $(2 \times H \times C)$. $\widehat{\chi}_3^*$ then undergoes convolution with a kernel size of $k \times k$, followed by batch normalization, resulting in a shape of $(1 \times H \times C)$. Finally, $\widehat{\chi}_3^*$ is passed through a Sigmoid activation layer to generate attention weights, which are then applied to $\widehat{\chi}_3$. Afterward, $\widehat{\chi}_3$ is rotated clockwise by 90° along the H -axis to maintain the same shape as the input χ .

Finally, the outputs of the three branches are averaged to obtain tensor y . This is represented by

$$y = \frac{1}{3} \left(\overline{\widehat{\chi}_1 \sigma(\psi_1(\widehat{\chi}_1^*))} + \chi \sigma(\psi_2(\widehat{\chi}_2)) + \overline{\widehat{\chi}_3 \sigma(\psi_3(\widehat{\chi}_3^*))} \right) \quad (8)$$

where σ is the Sigmoid activation function, and ψ_1, ψ_2, ψ_3 denote 2D convolutional layers with kernel size k .

The Z-pool layer is used to connect the operations of average pooling and maximum pooling to reduce the 0th dimension of the tensor to a tensor containing only two samples, preserving the rich features of the actual tensor and reducing the computational effort. Its equation is

$$\text{Z-pool}(\chi) = [\text{MaxPool}_{0d}(\chi), \text{AvgPool}_{0d}(\chi)] \quad (9)$$

Here, 0d denotes the 0-th dimension where the maximum pooling and average pooling operations occur.

4 Experiments and Analysis

4.1 Experimental Configuration

The experimental environment uses the Linux system Ubuntu18.04 with 14 vCPU Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz; RTX 3090GPU graphics card is used for model training and prediction with 24 GB of graphics memory. Python version 3.8 is used, and the deep learning framework Pytorch is the platform for model training. The COCO2017 dataset [34] was used for model training and validation, and the key point location was selected as the point with the largest prediction value offset by 1/4 to the next largest value, which is the final key point location. The initial learning rate was set to 0.001, the decay multiplier was 0.1, and the learning rate decayed between 170 and 200 rounds. The total number of iterations is 210 rounds.

The images in the dataset were preprocessed before training by cropping the image size of the COCO2017 dataset to 256×192 . Data enhancement operations were performed on the input images, including random rotation of the data (between $-45 - 45$ degrees), random horizontal flipping, and random scaling to improve the robustness of the model to the images.

4.2 Dataset and Evaluation Indicators

The COCO2017 dataset comprises image data that showcases a wide variety of human poses, scale variations, and occlusion patterns. It consists of three parts: the training set, the validation set, and the test set. In this paper, the proposed method trains the model on the training set, which consists of a total of 57 000 images and 150 000 instances of human beings. Each human instance is labeled with 17 key points, as shown in Tab. 2. The validation set contains 5 000 images to evaluate the effectiveness of the proposed model.

In the COCO2017 dataset, the evaluation metrics employed were object keypoint similar-

Tab. 2 Categories of key points

ID	Category	ID	Category
0	Nose	9	left wrist
1	Left eye	10	right wrist
2	Right eye	11	Left hip
3	Left ear	12	Right hip
4	Right ear	13	Left knee
5	Left shoulder	14	Right knee
6	Right shoulder	15	Left ankle
7	Left elbow	16	Right ankle
8	Right elbow		

ity (OKS), average precision (AP), and Average recall (AR). OKS, a standard metric in human keypoint detection tasks, signifies the level of similarity between the predicted and actual keypoints. It ranges from 0 to 1, with higher values indicating greater similarity. Its formula is defined as

$$\text{OKS} = \frac{\sum_i e^{-d_i^2/2s^2k_i^2} \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \quad (10)$$

where i represents the i th keypoint marker bit of the human body, v_i represents the visibility of the i th keypoint marker, δ refers to the value of 1 when the keypoint marker is visible and 0 otherwise, d_i is the euclidean distance value between the target keypoint and the labeled keypoints, s refers to the square root of the target area, and k_i represents the value of the attenuation constant for the keypoint category i .

4.3 Experimental Verification Analysis

The AT-HRNet model was trained and validated on the COCO2017 dataset, and its perfor-

mance was experimentally compared with other pose estimation network models. Tab. 3 presents a comparison of the experimental results on the COCO2017 validation set.

Tab. 3 demonstrates that AT-HRNet has shown improvements in mAP compared to Hourglass and CPN by 9.7% and 8.0%, respectively. Furthermore, compared to SimpleBaseline (ResNet-152), GNNet, and ENNet, AT-HRNet has enhanced mAP by 4.6%, 1.8%, and 0.6%, respectively. Additionally, relative to SimpleBaseline (ResNet-152) and ENNet, AT-HRNet has increased AR by 1.7% and 0.6%, respectively. Compared to the HRNet-32 method, AT-HRNet shows improvements in mAP by 3.2%, AP⁵⁰ by 3.9%, AP⁷⁵ by 4.0%, AP^M by 3.9%, AP^L by 1.6%, and AR by 0.6%. Analyzing from the perspective of model parameters, AT-HRNet shows an increase of 1.1 M parameters compared to HRNet. Despite the increase in model parameters and computational complexity, improvements are evident in other performance aspects, significantly enhancing the final predictive effectiveness.

Analysis of the experimental results reveals that, compared to HRNet, AT-HRNet primarily improves on the AP⁷⁵ and AP^M evaluation metrics. AP⁷⁵ represents the accuracy of the model when OKS = 0.75, demanding higher precision in terms of Euclidean distance compared to OKS = 0.50, bringing it closer to the true positions of key points. Thus, the improvement in this evaluation metric indicates that TripNeck and Trip-

Tab. 3 Comparison of Results from Different Methods in COCO2017 VAL

Method	Backbone	Input Size	Params/ 10^7	GFLOPs	mAP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
Hourglass	Hourglass	256×192	2.51	14.3	66.9	—	—	—	—	—
CPN	ResNet-50	256×192	2.70	6.2	68.6	—	—	—	—	—
SimpleBaseline	ResNet-50	256×192	3.40	8.90	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline	ResNet-101	256×192	5.30	12.4	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline	ResNet-152	256×192	6.86	15.7	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-32	HRNet-32	256×192	2.85	7.10	73.4	89.5	80.7	70.2	80.1	78.9
GNNet	HRNet-32	256×192	2.92	7.27	74.8	90.6	82.2	71.1	81.6	80.1
ENNet	HRNet-32	256×192	2.90	8.20	76.0	93.6	83.7	73.3	83.5	78.9
AT-HRNet	HRNet-32	256×192	2.96	7.31	76.6	93.4	84.7	74.1	81.7	79.5

Block effectively enhance the model’s performance in high-precision keypoint detection. AP^M indicates detection accuracy for medium-sized objects. It is observed that AT-HRNet also exhibits notable improvements in detecting medium-sized objects. Therefore, the method proposed in this paper has a greater advantage in detecting human key points.

4.4 Ablation Experiment

To validate the effectiveness of each module in AT-HRNet, we tested it on the COCO2017 validation set by adding the designed modules one by one. Throughout the process, the model was continuously optimized by tuning the parameters. The analysis of the results from the ablation experiments in Tab. 4 reveals that the introduction of each module has significantly enhanced the performance of AT-HRNet.

Tab. 4 Ablation Experiments

Method	Model			Params/ 10^7	GFLOPs	mAP
	AcBlock	TripNeck	TripBlock			
HRNet				2.85	7.10	73.4
	√			2.85	7.17	74.3
		√		2.89	7.15	74.7
Ours			√	2.90	7.19	75.2
	√	√		2.89	7.22	75.4
	√	√	√	2.96	7.31	76.6

Notably, the inclusion of the AcBlock module introduces almost no additional parameters while improving the average accuracy by 0.9% compared to HRNet. When only the TripNeck module is added, the network’s parameter count increases by 4.0×10^5 , resulting in a 1.3% improvement in average accuracy compared to HRNet. The introduction of both the AcBlock and TripNeck modules yields a 2.0% improvement in average accuracy compared to HRNet. When only introducing the TripBlock module, although the number of parameters in the network increased by 5.0×10^5 and the computational load increased by 0.09 GFLOPs, the average accuracy compared to HRNet improved by 1.8%. Finally, with the simultaneous introduc-

tion of three modules, AT-HRNet achieves a substantial 3.2% improvement in average accuracy compared to HRNet.

Based on the aforementioned experimental results, it is evident that the incorporation of AcBlock enhances the network’s capability to adaptively extract crucial information from diverse regions. This improvement effectively enhances the channel information extraction ability of ordinary convolution. The designed TripNeck and TripBlock enable the network to simultaneously capture local and global semantic information during cross-dimensional interactions, effectively improving the impact of irrelevant information when directly fusing multi-resolution branches.

To validate the efficiency of the TripBlock module in AT-HRNet, experiments were conducted on the COCO2017 dataset by introducing mainstream SENet and CBAM attention mechanisms. The results are shown in Tab. 5. It indicates that, compared to HRNet, the introduction of the TripBlock module has led to an average accuracy improvement of 1.8%. While accuracy also improves with the introduction of SENet and CBAM, their effects are not as pronounced as those observed with TripBlock. Based on the above analysis, the performance improvement of the SENet attention mechanism is relatively small. This is because it only enhances the model’s focus on channels without considering spatial features. Although CBAM addresses the feature relationships between channels and spatial dimensions, its average accuracy is still lower than that of TripBlock. TripBlock can perform cross-dimensional feature interactions between

Tab. 5 Comparative Experiments

Method	Model			Params/ 10^7	GFLOPs	mAP
	SENet	CBAM	TripBlock			
				2.85	7.10	73.4
Ours	√			2.85	7.11	74.1
		√		2.87	7.12	74.5
			√	2.90	7.19	75.2

channels and spatial dimensions, thereby enhancing the model’s ability to extract important features and reducing the loss of crucial information.

4.5 Visualization

Fig. 7 displays the detection results of partial images from the COCO2017 dataset under the AT-HRNet model and compares them with the visualization of HRNet detection results. Each human body has 17 target keypoints, which are connected by lines. Fig. 7(a)–(d) show the visualization results of different scenarios in single-person pose estimation. In Fig. 7(a), both models accurately locate keypoints in unoccluded scenarios. However, in scenes with occlusion, varying lighting, and small targets, AT-HRNet achieves more accurate detection than HRNet, as seen in the left eye of the person in Fig. 7(b), the legs in Fig. 7(c), and the shoulders in Fig. 7(d).

Fig. 7(e)–(h) depict the visualization results of different scenarios in multi-person pose estimation. Similarly, in unoccluded scenarios, both methods perform well, as shown in Fig. 7(e). However, in some complex poses, environments with occlusion, and varying lighting scenarios, AT-HRNet exhibits superior and more advantageous detection performance compared to HRNet, as demonstrated in Fig. 7(f)–(h). In summary, the proposed AT-HRNet model captures more important features compared to HRNet, including accurately detecting the positions of target keypoints, achieving higher detection accuracy, and possessing certain robustness and anti-interference capabilities.

5 Conclusion

To improve the model performance of human

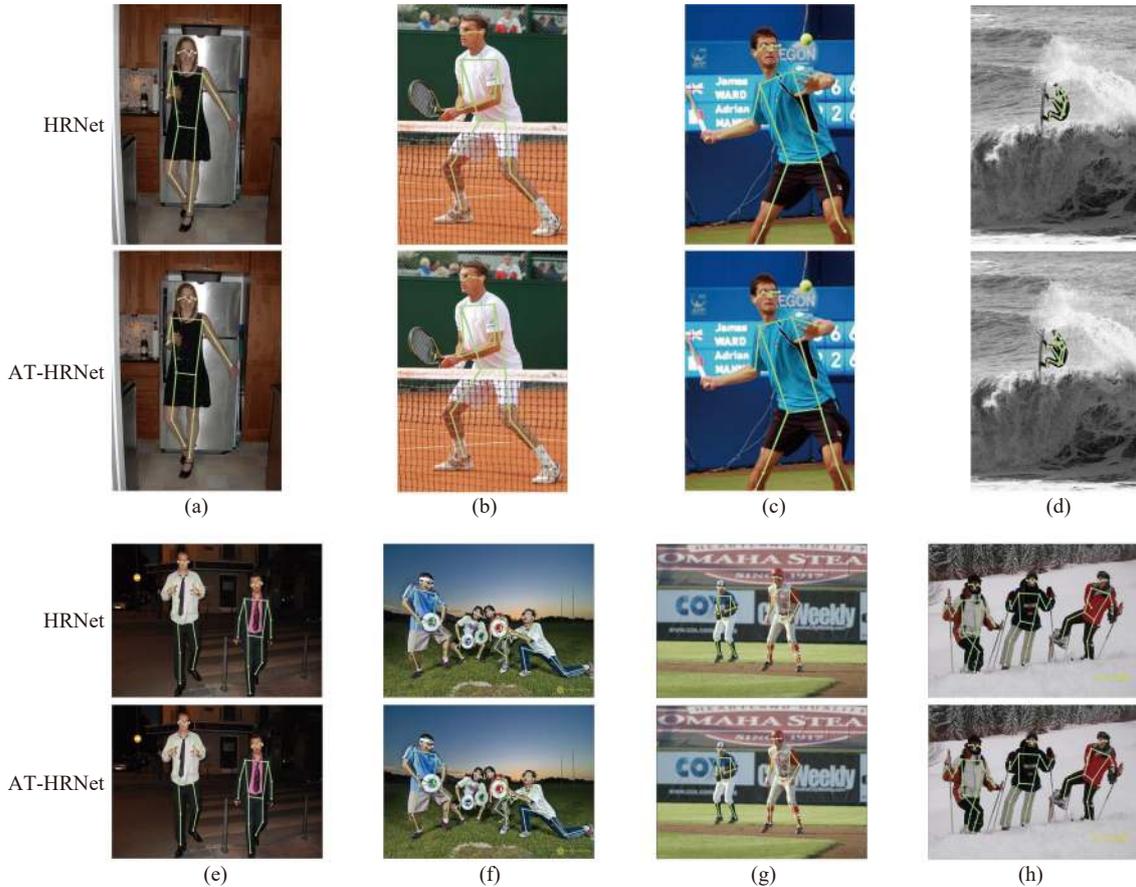


Fig. 7 Visualization of model detection effect: (a) single-person unoccluded; (b) single-person occlusion; (c) single-person good lighting; (d) single-person poor lighting; (e) multi-person unoccluded; (f) multi-person occlusion; (g) multi-person good lighting; (h) multi-person poor lighting

pose estimation models, AT-HRNet is proposed, which is a network that fuses convolutional self-attention and cross-dimensional feature transformations, with HRNet being a baseline model. The AcBlock is designed to leverage the strengths of both convolution and self-attention, focusing on important feature information in different regions and aggregating it within local receptive domains. This enhances the model's ability to extract feature maps effectively. The designs of TripNeck and TripBlock further improve the model's capability to extract important features during training. Experimental results on the COCO2017 validation set demonstrate that AT-HRNet achieves a 3.2% improvement in mAP compared to HRNet, exhibiting enhanced robustness and accuracy across different environments. Future work will further explore how to reduce model computational complexity while improving accuracy and apply human pose estimation to more domains.

References:

- [1] E. Martini, M. Boldo, S. Aldegheri, N. Valè, M. Filippetti, N. Smania, M. Bertuccio, A. Picelli, and N. Bombieri, "Enabling gait analysis in the telemedicine practice through portable and accurate 3D human pose estimation," *Computer Methods and Programs in Biomedicine*, vol. 225, pp. 107016, 2022.
- [2] J. Jiang, W. Skalli, A. Siadat, and L. Gajny, "Effect of face blurring on human pose estimation: ensuring subject privacy for medical and occupational health applications," *Sensors*, vol. 22, no. 23, pp. 9376, 2022.
- [3] L. D. Haberkamp, M. C. Garcia, and D. M. Bazett-Jones, "Validity of an artificial intelligence, human pose estimation model for measuring single-leg squat kinematics," *Journal of Biomechanics*, vol. 144, pp. 111333, 2022.
- [4] D. Kim, H. Park, T. Kim, W. Kim, and J. Paik, "Real-time driver monitoring system with facial landmark-based eye closure detection and head pose recognition," *Scientific Reports*, vol. 13, no. 1, pp. 18264, 2023.
- [5] A. Aafreen, A. R. Khan, A. Khan, A. Ahmad, M. A. Shaphe, A. Alzahrani, A. Alhusayni, A. H. Alameer, and R. A. Alajam, "Decoding the impact of driving postures: comparing neck pain, mobility, proprioception in car and bike drivers with and without forward head posture," *Journal of Transport & Health*, vol. 33, pp. 101719, 2023.
- [6] N. Fang, L. Qiu, S. Zhang, Z. Wang, Y. Gu, and K. Hu, "The rapid construction method of human body model for virtual try-on on mobile terminal based on MDD-Net," *Soft Computing*, vol. 26, no. 22, pp. 12023-12039, 2022.
- [7] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55-79, 2005.
- [8] R. C. Luo and S. Y. Chen, "Human pose estimation in 3-D space using adaptive control law with point-cloud-based limb regression approach," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 51-58, 2015.
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, pp. 886-893, 2005.
- [10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [11] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR 2011. IEEE*, Colorado Springs, CO, USA, pp. 1385-1392, 2011.
- [12] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, NSW, Australia, pp. 3487-3494, 2013.
- [13] K. YOSHIMURA and K. ICHIGE, "Human pose estimation by convolutional neural network and truncated nuclear norm minimization," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. J103-A, no. 7, pp. 152-155, 2020.
- [14] J. X. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology*, vol. 5, no. 23, pp. 495, 2017.

- [15] Z. W. Li, F. Liu, W. J. Yang, S. H. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, 2021.
- [16] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: convolutional triplet attention module," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, pp. 3139-3148, 2021.
- [17] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 815-825, 2022.
- [18] A. Toshev and C. Szegedy, "DeepPose: human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 1653-1660, 2014.
- [19] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 4724-4732, 2016.
- [20] A. Newell, K. Yang, and J. Deng, "Stacked hour-glass networks for human pose estimation," in *Computer Vision—ECCV 2016: 14th European Conference*, Amsterdam, The Netherlands, pp. 483-499, 2016.
- [21] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Real-time multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7291-7299, 2017.
- [22] B. Xiao, H. Wu and Y. Wei. "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 466-481, 2018.
- [23] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 5693-5703, 2019.
- [24] B. Cheng, B. Xiao, J. D. Wang, H. H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 5385-5394, 2020.
- [25] C. Q. Yu, B. Xiao, C. X. Gao, L. Yuan, L. Zhang, N. Sang, and J. D. Wang, "Lite-hrnet: A lightweight high-resolution network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 10435-10445, 2021.
- [26] Y. Zhang, Y. R. Huang, and P. K. Liu, "Research on multi-resolution human pose estimation with attention mechanism," *Computer Engineering and Applications*, vol. 57, no. 8, pp. 126-132, 2021.
- [27] M. S. Luo, Y. Xu, and X. X. Ye, "Human pose estimation using high resolution network with dual attention," *Computer Engineering*, vol. 48, no. 2, pp. 314-320, 2022.
- [28] Y. Zhang, G. Z. Wen, S. Y. Mi, M. L. Zhang, and X. Geng, "Overview on 2D human pose estimation based on deep learning," *Journal of Software*, vol. 33, no. 11, pp. 4173-4191, 2022.
- [29] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 1831-1840, 2017.
- [30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7132-7141, 2018.
- [31] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 3-19, 2018.
- [32] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7794-7803, 2018.

- [33] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: non-local networks meet squeeze-excitation networks and beyond, " in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul, Korea (South), pp. 1971-1980, 2019.
- [34] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: common objects in context, " in *Computer Vision—ECCV 2014: 13th European Conference*, Zurich, Switzerland, pp. 740-755, 2014.



Anzhan Liu is presently an associate professor in Zhongyuan University of Technology, Zhengzhou, China. He received the Master's degree in Computer Software and Theory from Huazhong University of Science and Technology in 2008. His research focuses on the computer vision, machine and deep learning, intel-

ligent computing, etc.



Yilu Ding received the B.E. degree from North China University of Water Resources and Electric Power of Computer Science and Technology, China, in 2022. She is currently studying toward the M.S. degree in Zhongyuan University of Technology. Her research interests include computer vision and human pose estimation.



Xiangyang Lu is presently an associate professor in Zhongyuan University of Technology, Zhengzhou, China. He received the M.S. degree in college of mechanical and Electrical Engineering from East China University of Technology in 2005, and the Ph.D. degree in Electronic and Information from Beijing Institute of Technology in 2014. His research focuses on the information security operation, deep learning and computer vision.