applied sciences

MDPI

Article

An Improved High-Resolution Network-Based Method for Yoga-Pose Estimation

Jianrong Li, Dandan Zhang, Lei Shi, Ting Ke and Chuanlei Zhang *

College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin 300453, China; lisa_ljr@tust.edu.cn (J.L.); zdd15076311103@163.com (D.Z.); shil0406@163.com (L.S.); keting@tust.edu.cn (T.K.)
* Correspondence: 97313114@tust.edu.cn; Tel.: +86-13-9117-8065-3

Abstract: In this paper, SEPAM_HRNet, a high-resolution pose-estimation model that incorporates the squeeze-and-excitation and pixel-attention-mask (SEPAM) module is proposed. Feature pyramid extraction, channel attention, and pixel-attention masks are integrated into the SEPAM module, resulting in improved model performance. The construction of the model involves replacing ordinary convolutions with the plug-and-play SEPAM module, which leads to the creation of the SEPAMneck module and SEPAMblock module. To evaluate the model's performance, the YOGA2022 human yoga poses teaching dataset is presented. This dataset comprises 15,350 images that capture ten basic yoga pose types—Warrior I Pose, Warrior II Pose, Bridge Pose, Downward Dog Pose, Flat Pose, Inclined Plank Pose, Seated Pose, Triangle Pose, Phantom Chair Pose, and Goddess Pose—with a total of five participants. The YOGA2022 dataset serves as a benchmark for evaluating the accuracy of the human pose-estimation model. The experimental results demonstrated that the SEPAM_HRNet model achieved improved accuracy in predicting human keypoints on both the common objects in context (COCO) calibration set and the YOGA2022 calibration set, compared to other state-of-the-art human pose-estimation models with the same image resolution and environment configuration. These findings emphasize the superior performance of the SEPAM_HRNet model.

Keywords: human pose estimation; attention mechanism; high-resolution networks; feature pyramids

Citation: Li, J.; Zhang, D.; Shi, L.; Ke, T.; Zhang, C. An Improved High-Resolution Network-Based Method for Yoga-Pose Estimation. *Appl. Sci.* **2023**, *13*, 8912. https://doi.org/10.3390/app13158912

Academic Editor: Douglas O'Shaughnessy

Received: 9 June 2023
Revised: 26 July 2023
Accepted: 31 July 2023
Published: 2 August 2023

1. Introduction

Two dimensional (2D) human pose estimation has been a hotspot of research in the computer vision field because of its characteristics. Its task is to predict and localize the keypoint parts of a target human body. This is the basis of other action tasks, and 2D human pose estimation has important research and application values. The evaluation of human motion action is receiving more and more attention in the computer vision field, and human pose estimation, as the basis of human action evaluation tasks, has also been the subject of much research, with achieved results.

Based on the excellent performance and advancement of deep convolutional neural networks, Toshev et al. [1] (2014) introduced deep neural networks into human pose-estimation algorithms for the first time. The model considered the localization of keypoints as a regression problem of human keypoints, extracted image features by convolutional neural networks, and simultaneously modeled the relationship between keypoints using convolutional kernels, which improved the performance of predicting human keypoints.

A stacked hourglass network [2] uses a symmetric modular structure from high-resolution downsampling to low-resolution downsampling, followed by low-resolution upsampling to recover the high resolution to fuse the features of different scales to capture multiple spatial relationships. This results in the improvement of the accuracy of the human body keypoints prediction.

On this basis, a simple baseline method [3] simplifies the tandem structure from high resolution to low resolution and, then, recovers high resolution, generating high-resolution feature representations from low-resolution feature representations through

three transposed convolutional layers. The HRNet high-resolution network proposed in 2019 [4] can be distinguished from the previous tandem approach (from high resolution to low resolution, then recovering to high resolution). The HRNet model adopts the parallel connection of high resolution and low resolution to maintain the high-resolution feature map throughout the model, which compensates for the lack of spatial resolution loss generated by upsampling.

However, the human body is a non-rigid structure that produces a wide variety of postures and is prone to self-obscuration problems that interfere with the accurate localization of human keypoints. To improve the accurate localization of occluded keypoints, more attention needs to be paid to these regions. Fortunately, model performance can be significantly improved by embedding the attention module into existing CNNs, enabling the network to pay more attention to these key regions. By learning to redistribute the channel weights of convolutional features, the attention mechanism improves model performance, with a slight increase in computational complexity, leading to more accurate prediction of keypoints in the human body.

Currently, attention mechanisms such as SENet [5], CBAM [6], and GCNet [7] have achieved sizable performance improvements. Nevertheless, these attention mechanisms can only capture local information, making it difficult to establish long-term dependencies, and they cannot fully access and utilize the spatial information of feature maps at different scales or enrich the feature space. To solve this problem, this paper proposes the squeeze-and-excitation and pixel-attention-mask (SEPAM) module, which utilizes the feature pyramid structure [8] to extract features at different scales using convolution kernels of different sizes. The weights are assigned to different channels by the channel attention (SE) module; then, the pixel positions are assigned by the pixel-attention-mask (PAM) module. Used in parallel, these two attention modules can effectively extract finer-grained multi-scale spatial information, while establishing channel dependencies over longer distances and pinpointing feature information in the spatial direction, thus reducing the loss associated with keypoint localization. This optimization helps improve the accuracy of the human pose-estimation model in predicting the keypoints of the human body, especially the keypoints of the occluded parts.

Based on the human posture estimation model named SEPAM_HRNet, which was formed based on the application of the SEPAM module to the HRNet network, we created a human yoga movement posture feature dataset named YOGA2022. This dataset was constructed with labels for ten common human yoga movement postures. It provides the location coordinates of, as well as visibility information about, 17 human keypoints that are consistent with the label form of the COCO dataset. With this new dataset, a deeper understanding of the postural characteristics of human yoga poses can be gained, and the accuracy and stability of the SEPAM_HRNet model in human keypoint prediction can be further validated and optimized. Meanwhile, the establishment of this dataset provides a valuable resource and foundation for future research, which is expected to lead be more in-depth and to have applications in the field of human-motion analysis and posture estimation.

Overall, the contributions of the work in this paper are as follows:

1. Considering the specificity of yoga movement postures that are prone to self-obscuring phenomenon, this paper creates a new dataset of human yoga movement postures, with truth labels, called YOGA2022.
2. In order to solve the problem of self-obscuring keypoint prediction, which requires finer multi-scale spatial information, this paper proposes the SEPAM module and fuses it into the HRNet model to establish the new SEPAM_HRNet model. The SEPAM_HRNet model improves prediction accuracy without introducing a large number of parameters or complexity. Through this innovative fusion, the self-obscuration problem can be better dealt with.
3. After extensive comparative experiments, the experimental results fully validated the effectiveness of the lightweight channel spatial pyramid attention (SEPAM) module.

2. Related Work

2.1. Human Posture-Estimation Model

Before introducing 2D human pose-estimation methods by deep learning, the human pose problem was mainly solved using methods based on graph structure models [9,10]. However, these methods are greatly affected by factors such as figure occlusion, shooting angle, and image illumination, with limited representation capability and poor model prediction accuracy, which is challenging in meeting the needs of practical applications.

With the continuous development of deep-learning network models, the performance of human posture estimation has been significantly improved. This paper briefly reviews the deep-learning-based pose estimation methods in recent years. Deep-learning-based 2D human pose =estimation methods can be categorized into top-down and bottom-up architectures. The top-down architecture first detects a single-person image region using an excellent human target detector [11,12], then uses a human pose estimator [2–4,13] to localize the human keypoints in each box. In contrast, the bottom-up human pose-estimation method [14–17] first detects all the human keypoints in the image, then obtains the pose of each person by matching. The bottom-up method has faster operation speed and higher real-time performance thano the top-down method, but it is prone to errors for the same types of keypoints that are close to each other.

Two-dimensional human pose estimation [2–4,17–20] has dominated the field of human pose estimation in terms of performance. Several studies [4,18–23] constructed new pose estimation network architectures to extract better features. The creation of the HRNet [4,16] family of models was a notable development in the area of pose recognition. The HRNet model employed a new convolutional neural network (CNN) architecture designed to model high-resolution feature responses, demonstrating the use of the 2D human pose estimation. YOLO-Pose [18] performed keypoint regression and grouping by utilizing the centroid of the bounding box for yolo target detection as the initial point to achieve accurate localization of the keypoints. TDMI [22] proposed a mutual information-based time-difference learning model that utilized the mutual information to direct the model to learn the features relevant to the task, improving the performance of human pose estimation via video. PoseIG [23] designed attribute-based metrics to assist in analyzing and diagnosing pose-estimation frameworks. Other studies [24–29] built on the optimization perspective and tried to reduce the inference latency of pose estimation by lightweighting the modules or reducing the network branches, but often at the cost of drastically reducing the accuracy of pose estimation. The optimization perspective also includes studies that employed knowledge distillation methods [30–32] by constructing lightweight mini-models and using supervisory information from larger models with better performance to train the mini-models, with a view to obtaining better performance and accuracy. For example, OKDHP [30] proposed an online positional distillation method to extract positional structure knowledge in a single-stage approach. DistilPose [32] proposed a heatmap-regression distillation framework to achieve knowledge transfer between heatmap-based and regression methods.

In recent research, Transformer has achieved great success in the field of natural language processing (NLP). In the field of 2D human pose estimation, many studies [19,20,33–36] have also incorporated Transformer structures. TFPose [33] first introduced Transformer into a pose estimation framework in the form of regression; PRTR [35] proposed a two-stage end-to-end regression based on a cascaded Transformer framework that achieved excellent performance in regression-based approaches; TransPose [20] and TokenPose [19] introduced Transformer for heat map-based human pose estimation and achieved comparable performance. However, these Transformer-based pose estimation models did not perform as well as CNNs in obtaining localized information and had a large computational overhead, which made them difficult for practical applications. Therefore, in this paper, the focus is on HRNet, which represents the state-of-the-art performance among CNN networks, as the base network for the research.

2.2. Attentional Mechanisms

In the field of image processing, the attention mechanism is widely used to improve the effectiveness of models. These mechanisms redistribute channel weights in feature maps obtained from convolutional neural networks to perform functions such as squeeze, excitation, and attend. Studies have shown that incorporating attention mechanisms can enhance the performance of lightweight models with only a slight increase in computational complexity. One popular attention module is the squeeze-and-excitation (SE) attention module [5]. This module focuses on modeling the relationship between channels through two main operations: squeeze and excitation. In the squeeze phase, it performs feature compression in the spatial dimension through a global average pooling operation. Then, in the excitation phase, it generates weight parameters for each channel to determine its importance by using a fully connected layer and a nonlinear activation function, which are weighted by multiplying these weights with the original feature map, thus modeling the relationship between feature channels and completing the recalibration in the channel dimension of the original feature map.

SKNet [37] is an improvement over SENet that performs complete convolution operations on input feature maps using two different kernel sizes. The resulting convolved feature maps are then summed, followed by a global pooling operation. Subsequently, two fully connected layers are used to downscale and upscale the output, obtaining two attention coefficient vectors, denoted as "a" and "b". These vectors are individually weighted with the two previous feature matrices, and the weighted feature maps are summed.

However, these attention modules do not consider the significance of spatial orientation information. To address this limitation, the convolutional block attention module (CBAM) [6] was proposed. CBAM takes into account both channel and spatial relations and generates a separate attention map for spatial attention. By multiplying this attention map with the input feature maps, the resulting feature maps incorporate attention weights. CBAM effectively captures spatial direction information in the attention map, while preserving the understanding of channel relationships. Additionally, the pyramid attention segmentation module was introduced to extract finer-grained multi-scale spatial information and to establish longer-range channel dependencies. This allowed for the learning of richer multi-scale feature representations and adaptive feature recalibration across multiple dimensions by dynamically adjusting channel-attention weights [38].

3. Structure of the SEPAM_HRNet Model

The SEPAM_HRNet model proposed in this paper inherits the basic four-stage architecture of the original HRNet [4] network, and the model architecture diagram is shown in Figure 1.

The SEPAM_HRNet model first passes through two convolutional layers that are downsampled on the basis of the original image, then passes through the Stage 1 phase, which consists of four SEPAMneck modules. After that, a staged progressive downsampling phase is performed, which consists of three stages—Stage 2, Stage 3, and Stage 4—with each branch of each stage consisting of four SEPAMblock modules. Gradually decreasing the resolution of the feature maps can reduce the loss of detailed features in the human posture feature maps caused by the large downsampling operation. The use of parallel branches with different resolutions and channel numbers in each stage avoids the loss of spatial information in the feature map during the downsampling process.

The specific processing of the SEPAM_HRNet model is as follows:

1. The input image is first pre-processed by two convolutional layers with a kernel size of 3×3 and a step size of 2 (including BN and RELU operations). The input feature map's resolution is then downsampled by a factor of 4 to become 1/4 of the original, and the number of channels is increased from three channels to 64 channels.
2. The pre-processed feature map is sent into the Stage 1 stage, which is made up of four SEPAMneck modules, where the shallow features are removed from the feature map,

while only the feature map's channel count is altered and the feature layer's size is left unchanged.

3. In Stages 2, 3, and 4, downsampling operations are performed progressively. The feature maps used in each stage have resolutions of 1/4, 1/8, 1/16, and 1/32 of the original resolution, corresponding to channel numbers C, 2C, 3C, and 4C, respectively. Each branch in each stage consists of four SEPAMblock modules. At the end of each stage, upsampling/downsampling is used to unify feature maps of different resolutions to the same size and perform element-wise addition for multi-scale feature fusion. Finally, only the output of the first high-resolution branch is subjected to a 1 × 1 convolution to obtain 17 keypoints heatmaps, which completes the keypoints prediction.

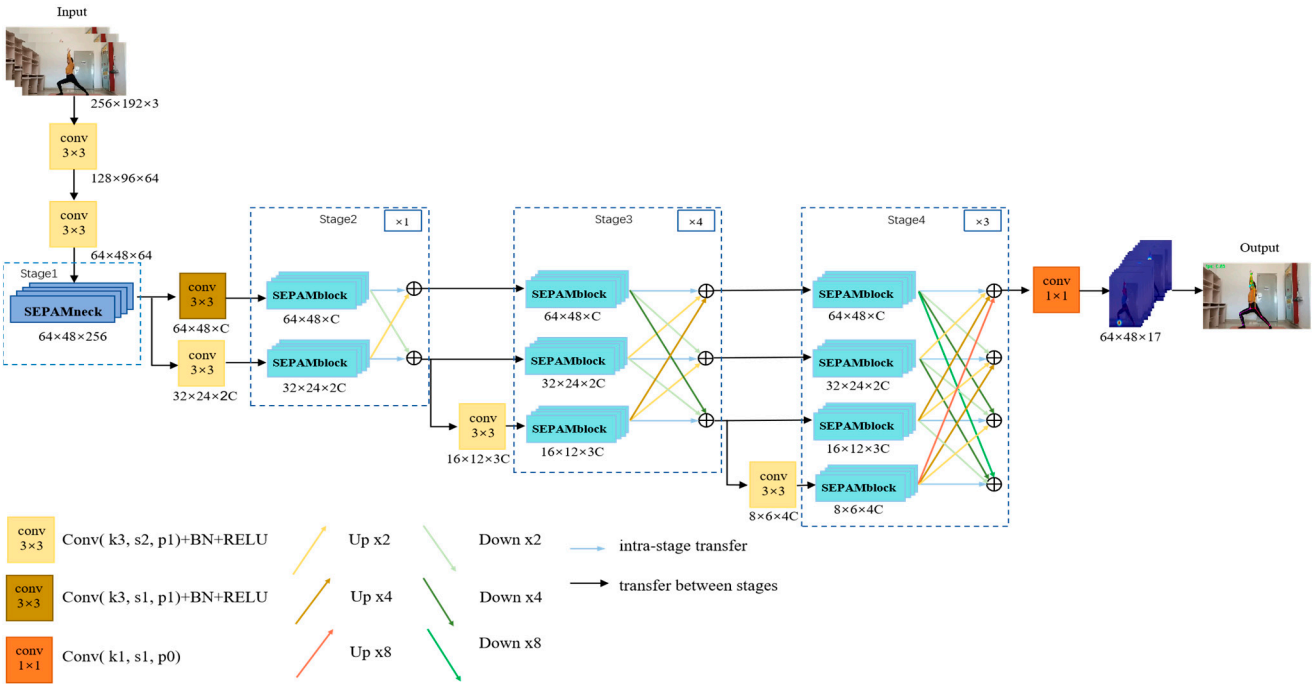


Figure 1. Structure of SEPAM_HRNet model. SEPAM_HRNe starts with a high-resolution branch and gradually adds high-resolution streams to low-resolution streams as the main body. The main body has a series of stages, each containing parallel multi-resolution streams and repeated multi-resolution fusion.

In this paper, the SEPAM_HRNet model architecture is adopted, with the number of channels C = 32 (i.e., the number of feature map channels in the branch where the maximum resolution feature map is 32). As the stage progresses, the resolution and the number of channels of the feature map is adjusted for each stage. In this paper, referring to the idea of HRNet [4] literature, for every 2X downsampling branch, the number of feature map channels in the added branch is doubled, thus compensating for the loss of spatial localization caused by resolution degradation. The specific implementation methods of the SEPAMneck module and SEPAMblock module structures are described in the summary provided in Section 3.2.

3.1. SEPAM Module

Existing attention modules have limitations in capturing global information, establishing long-term information dependencies, and effectively utilizing information at different scales in feature space. To overcome these limitations, this paper proposes a lightweight squeeze-and-excitation and pixel-attention-mask (SEPAM) module, which can capture precise location information and areas of interest in the spatial direction, while obtaining

feature information between channels. The SEPAM module sequentially implements three functions of feature extraction at different scales, the channel attention module and the pixel-attention-mask module. This has important implications for the network's feature representation and task-performance improvement. The specific structure of the SEPAM module is shown in Figure 2.

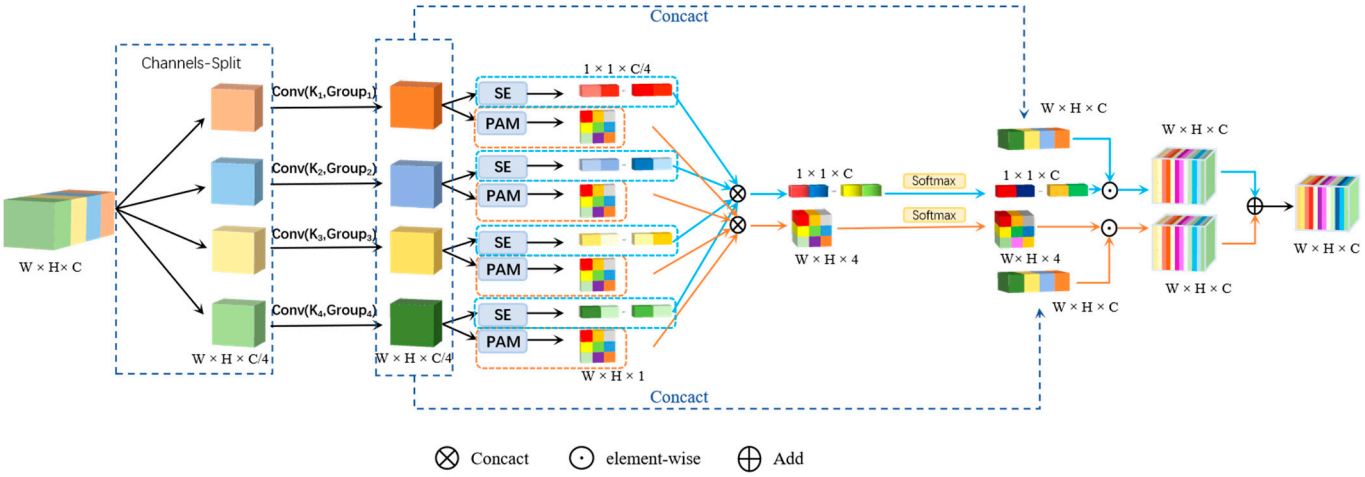


Figure 2. SEPAM module structure. Blue indicates the channel attention (SE) branch, and orange indicates the pixel-attention-mask (PAM) branch.

The steps for the SEPAM module are as follows.

1. Feature map split. The input feature map X is averaged in the channel dimension to obtain four parts of the feature map, X_1, X_2, X_3, and X_4. Each part of the feature map X_i maintains the same resolution, and the number of channels becomes one-fourth of the original number of channels.
2. Multi-scale feature map acquisition. Features of various scales are retrieved for the feature maps X_i (i = 1, 2, 3, and 4) of the divided four branches using various convolutional kernel sizes K_i (i = 1, 2, 3, and 4), resulting in four feature maps of the same size and with the same number of channels, C/4. To deal with the input tensor at various kernel scales without increasing the computational cost, a group convolution operation is used for the feature maps of the four divided branches. The group size is computed as $Group_i = 2\exp((K_i - 1)/2))$, according to the size of the convolution kernel K_i for each branch, and these feature maps can be directly spliced in the convolution kernels of the other branches.
3. Generating attention weights. The feature maps generated from each branch in Step 2 are passed through the SE channel attention module and the pixel-attention-mask (PAM) module to obtain the attention weights in the channel and spatial pixel directions, respectively. According to the idea of CBAM [6], this paper also adopts the parallel connection of two attention modules.
4. Concatenating attention weights. The channel attention vectors and the pixel-attention-mask maps obtained in Step 3 are concatenated separately to realize the interaction of the attention information and the fusion of the cross-dimensional information, and then a softmax operation is performed to complete the recalibration of the channel-attention vectors and the pixel-attention-mask maps.
5. Output module results. The four feature maps obtained in Step 2 are spliced in the channel dimension to form a multi-scale feature map, X′. Then, the corrected attention vector and pixel-attention-mask map are applied to the feature map X′, respectively. Next, the two feature maps with attention weights are summed, and the result obtained is used as the final output of the SEPAM module.

According to the SE [5], the SE channel attention first performs a global average pooling operation (AvgPool) on the input feature map (W × H × C) to obtain a 1 × 1 × C feature vector. Then, it passes through two fully connected layers. The first fully connected layer reduces the number of neurons to C/r (r is the reduction factor), and the second fully connected layer increases the dimension to C neurons, then obtains channel weight values through the sigmoid function and performs element-wise with the original feature map, so that the neural network can focus on certain feature channels. The SE attention process is shown in Figure 3.

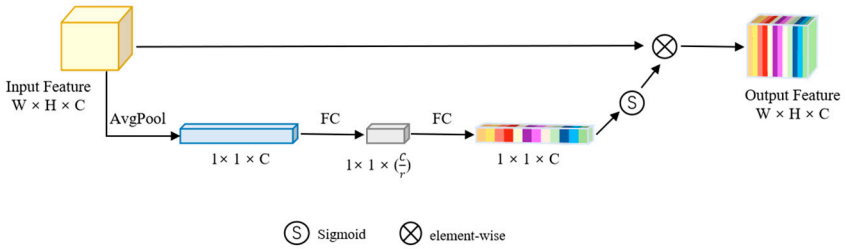


Figure 3. SE attention. A 1 × 1 × C weight matrix is obtained through a series of operations, where C denotes the number of channels; then, the original features are reconstructed (different colors denote different values to measure the importance of the channels).

The SE channel-attention module helps the network to boost proper feature channels and suppress less useful ones. However, on a 2D feature map, each pixel contains feature information. These items of pixel-level feature information vary in their degree of contribution to the task, and only task-relevant regions need to receive attention. In order to fully utilize the useful pixel-level feature information, this paper proposes a pixel-attention-mask (PAM) module that can generate a corresponding pixel-attention-mask map based on the input-feature map.

First, a W × H × C input-feature map is implemented through a 1 × 1 convolution to achieve a dimensionality reduction operation, and a W × H × (C/r) (r is a reduction factor) feature map is obtained. This process not only adds more nonlinear processing, but also fits complex correlations between channels. Next, a 3 × 3 convolution is used to further learn features, and a 1 × 1 convolution used to focus on the features compression obtains a W × H × 1 feature map. Then, the sigmoid layer is used to map the compressed one-dimensional features between (0, 1) and, finally, obtain a W × H × 1 pixel-attention-mask map, where the value of each pixel position represents the pixel at that position level of attention. Then, the element-wise operation is performed on the input-feature map of W × H × C and the pixel-attention-mask map of W × H × 1 to provide the feature map with pixel-level attention and to identify the features that need to be paid attention to in the image data. Useful features are increased and useless features are weakened to achieve the effect of feature screening and enhancement and to better learn critical features. The PAM attention-masking process is shown in Figure 4. This module can improve network performance and enable the network to understand and utilize pixel-level feature information more accurately.

The SEPAM integrates the functions of muti-scale feature extraction, channel attention, and the pixel-attention mask, so the number of parameters and complexity of operations are calculated as shown in Equations (1) and (2):

$$P_{SEPAM} = \sum_{i=1}^{4}[(K_i \times K_i \times \frac{C_{in}}{Group} \times \frac{C_{in}}{4Group} \times \frac{1}{Group}) + (\frac{C_{in}}{4} \times \frac{C_{in}}{4reduction} + \frac{C_{in}}{4reduction} \times \frac{C_{in}}{4}) + (1 \times 1 \times \frac{C_{in}}{4} \times \frac{C_{in}}{4reduction} + 3 \times 3 \times \frac{C_{in}}{4reduction} \times \frac{C_{in}}{4reduction} + 1 \times 1 \times \frac{C_{in}}{4reduction} \times 1)] \quad (1)$$

$$G_{SEPAM} = \sum_{i=1}^{4} H \times W \times [(K_i \times K_i \times \frac{C_{in}}{Group} \times \frac{C_{in}}{4Group} \times \frac{1}{Group}) + (\frac{C_{in}}{4} \times \frac{C_{in}}{4reduction} + \frac{C_{in}}{4reduction} \times \frac{C_{in}}{4}) + (1 \times 1 \times \frac{C_{in}}{4} \times \frac{C_{in}}{4reduction} + 3 \times 3 \times \frac{C_{in}}{4reduction} \times \frac{C_{in}}{4reduction} + 1 \times 1 \times \frac{C_{in}}{4reduction} \times 1)] \quad (2)$$

where K_i denotes the convolution kernel size; C_{in} denotes the input channel of the module; Group denotes the number of groups of grouping; reduction is the scaling factor; and H and W represent the height and width of the feature map, respectively. The SEPAM module can establish longer-distance channel dependencies in the network and provide different degrees of attention for pixel-level features through the combined application of the feature pyramid structure, SE channel attention, and the pixel-attention-mask map. It is worth noting that the HRNet model itself has four branches, and if a four-branch SEPAM structure is used in each branch of the SEPAMblock module, it will lead to too many branches in the whole model, which will aggravate the computational complexity of the model. Therefore, the feature map in SEPAMblock is not divided; instead, a 3 × 3 convolution operation is applied to the entire feature map.

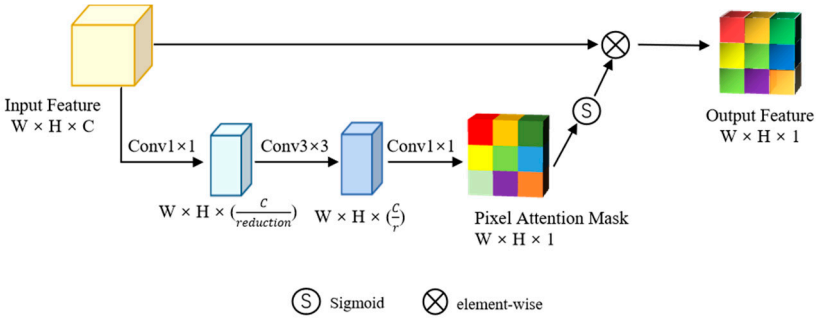


Figure 4. Pixel attention mask. A weight matrix of W × H × 1 is obtained through a series of operations, where W and H denote the width and height of the feature map, respectively; then, the original features are reconstructed (different colors denote different values to measure the importance of each pixel).

3.2. SEPAMneck and SEPAMblock Module Structure

The HRNet model mainly consists of a bottleneck module and a Basicblock module. The SEPAM_HRNnet model proposed in this paper redesigns the modules, based on the HRNet model. We propose the SEPAMneck module and the SEPAMblock module.

The structure of the SEPAMneck module is shown in Figure 5a. The overall structure is similar to that of the bottleneck module, which consists of three submodules on the main branch and a shortcut branch. The head and the tail of the SEPAMneck module are 1 × 1 convolutions, with a SEPAM module in the middle. Compared with the original bottleneck residual module, the SEPAM module proposed in this paper replaces the 3 × 3 convolution in the middle. The workflow of the SEPAMneck module is as follows: 1 × 1 convolution compresses the depth of the feature map, which is input to the SEPAM module to perform multi-scale feature extraction; then, 1 × 1 convolution is used to reduce the channel dimensions. The SEPAMneck module establishes the jump connections between the high-dimensional features. According to the idea of ResNet, jump connections are not added when the number of input and output channels are different.

The calculation formula of the parameters and operational complexity of the SEPAMneck module is shown in Equations (3) and (4):

$$P_{SEPAMneck} = 1 \times 1 \times C \times C_{in} + P_{SEPAM} + 1 \times 1 \times C_{in} \times C_{out} \quad (3)$$

$$G_{SEPAMneck} = H \times W \times (1 \times 1 \times C \times C_{in} + P_{SEPAM} + 1 \times 1 \times C_{in} \times C_{out}) \quad (4)$$

where C denotes the number of channels of the SEPAMneck module input-feature map; C_{in} is the number of channels of SEPAM module input; and C_{out} is the number of channels of the SEPAMneck odule input-feature map.

The structure of the SEPAMblock module is shown in Figure 5b. Compared to the original Basicblock module, the SEPAM module replaces the first 3 × 3 convolution to improve the feature map representation by redistributing the feature-map channels and

the pixel-position weights, then applies the standard 3 × 3 convolution to further extract features and preserve the hopping structure.

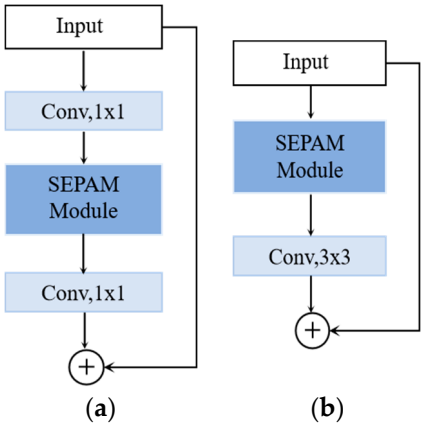


Figure 5. SEPAMneck module structure: (**a**) SEPAMblock module structure; (**b**) the SEPAMneck module is used to extract shallow features in the first stage and the SEPAMblock module is used in all subsequent branching stages.

The calculation formula of the parameters and the operational complexity of the SEPAMneck module are shown in Equations (5) and (6):

$$P_{SEPAMneck} = P_{SEPAM} + 3 \times 3 \times C_{in} \times C_{out} \quad (5)$$

$$G_{SEPAMneck} = H \times W \times (P_{SEPAM} + 3 \times 3 \times C_{in} \times C_{out}) \quad (6)$$

where C_{in} is the number of input channels to the SEPAM module and C_{out} is the number of channels of the SEPAMblock module input-feature map.

4. Experiments and Analysis of Results

4.1. Experimental Results for the COCO Dataset

4.1.1. Dataset Description

The common objects in context (COCO) data set is mainly used in computer vision tasks and contains more than 200,000 pictures and 250,000 personal instances. The data set annotation includes 17 keypoints: 0 for nose, 1 for left eye, 2 for right eye, 3 for left ear, 4 for right ear, 5 for left shoulder, 6 for right shoulder, 7 for left elbow, 8 for the right elbow, 9 for left wrist, 10 for right wrist, 11 for left hip, 12 for right hip, 13 for left knee, 14 for right knee, 15 for left ankle, 16 for right ankle. In this paper, the experiment trains the model on the COCO train2017 dataset (a total of 118,287 pictures) and evaluates the model on the val2017 dataset (a total of 5000 pictures).

4.1.2. Evaluation Metrics

For each human target, the true label of the keypoint is of the form [x_1, y_1, v_1,. . ., x_k, y_k, v_k], where x and y are the coordinates of the keypoints, and v is the visibility flag. The validation criteria for the experiments in this section are based on object keypoint similarity (OKS): AP^{50} is the accuracy of predicting keypoints when OKS = 0.5; AP^{75} is the accuracy of predicting keypoints when OKS = 0.75; the mean average precision (mAP) is AP (M), the accuracy of predicting keypoints for medium-sized objects; AP (L) is the accuracy of predicting keypoints for large-sized objects; and AR is the average of all predicted keypoints between 10 thresholds at OKS = 0.50, 0.55,. . . , 0.90, and 0.95. The average value of 10 threshold points is determined. The specific implementation is shown in Equation (7).

$$OKS = \frac{\sum_i \exp\left(-d_i^2/2s^2k_i^2\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \quad (7)$$

where d_i denotes the Euclidean distance between the detected keypoint and the true label corresponding to the keypoint in the dataset; v_i is the visibility flag bit of the true label of the keypoint; $v_i = 0$ means the keypoint is not labeled; $v_i = 1$ means the keypoint is marked but obscured; $v_i = 1$ means the keypoint is labeled and not obscured; for $\delta(x)$, the value is 1 when x is true and false when x is 0; s is the target scale, i.e., the segmented area of the target; and ki is the correlation constant that controls the decay of each keypoint. For each keypoint, a keypoint similarity in the range [0, 1] is generated. In the case of OKS = 1, there is a perfect prediction of the keypoint; in the case of OKS = 0, the predicted value is too far from the true value.

4.1.3. Experimental Configuration and Training Details

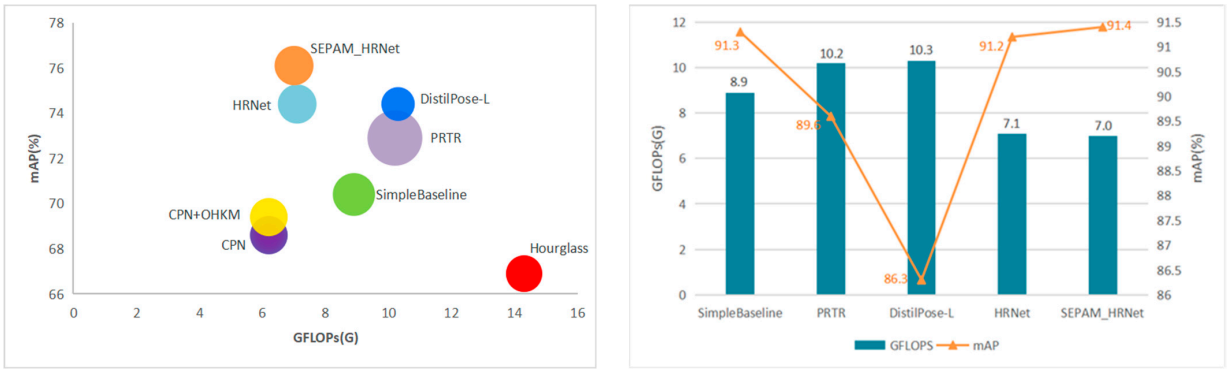
The experimental environment for this section is configured as follows: Ubuntu 18. 04 LST 64-bit system, a GeForce RTX 3090 graphics card, and the PyTorch 1.10.0 deep-learning framework. The training is performed on the COCO training set by cropping the images in the COCO training set and scaling them to a fixed 256 × 192. Adam is used as the optimizer for the network training, and the initial learning rate is 1×10^{-3}. The learning rate decays to 1×10^{-5} at round 210, for a total of 210 training rounds. The minimum batch size for each GPU is 32. The data are enhanced during training with random image rotation (between $-45°$ and 45°), random scaling (between 0.65 and 1.35), random horizontal flip, and half body (with a certain probability of cropping the target and keeping only half of the keypoints, upper or lower body). The loss function of the model, defined as the mean square error, is used to compare the predicted heatmap with the ground-truth heatmap. This loss function calculates the error at each pixel position between the predicted keypoint heatmap and the ground-truth keypoint heatmap and takes the average of these errors as the final loss.

4.1.4. Experimental Verification and Analysis

The experimental results of this section on the COCO calibration set are shown in Table 1. The results show that the SEPAM_HRNet model achieves higher accuracy than the original HRNet model with the same number of parameters as the HRNet model and with lower computational complexity. Compared with other recent human pose-estimation models, such as hourglass [2], CPN [39], CPN + OHKM [39], simple baseline [3], PRTR [35], and DistilPose-L [32], the SEPAM_HRNet model obtained 9.2, 7.5, 6.7, 5.7, 3.2, and 1.7 percentage points improvement on mAP, respectively. Compared to the HRNet [4] model, the SEPAM_HRNet model improved by 1.7 percentage points on mAP and 3.1 percentage points on AP50, while the other validation criteria remained comparable to the HRNet model. The experimental results are shown in Figure 6a.

Table 1. Performance comparison of COCO val2017. Comparison between SEPAM_HRNet and other methods on the COCO validation set. The notation "-" indicates that no reported results were available.

Models	Backbone	Input_Size	Params	GFLOPS	AP	AP.5	AP.75	AP(M)	AP(L)
Hourglass	Hourglass	256 × 192	25.1 M	14.3 G	66.9	-	-	-	-
CPN	ResNet-50	256 × 192	27.0 M	**6.2 G**	68.6	-	-	-	-
CPN + OHKM		256 × 192	27.0 M	**6.2 G**	69.4	-	-	-	-
Simple baseline		256 × 192	34 M	8.9 G	70.4	88.6	78.3	67.1	77.2
PRTR	HRNet	256 × 192	57.2 M	10.2 G	72.9	-	-	-	-
PRTR		384 × 288	57.2 M	21.6 G	73.1	89.4	79.8	68.8	80.4
DistilPose-L		256 × 192	**21.3 M**	10.3 G	74.4	89.9	81.3	71.0	81.8
HRNet	HRNet	256 × 192	28.5 M	7.1 G	74.4	90.5	81.9	70.8	81.0
HRNet		384 × 288	28.5 M	16.0 G	75.8	90.6	82.5	72.0	82.7
SEPAM_HRNet		256 × 192	28.8 M	7.0 G	**76.1**	**93.6**	**83.7**	**73.3**	80.1
SEPAM_HRNet		384 × 288	28.8 M	15.8 G	**77.6**	**93.6**	**84.7**	**74.7**	81.8



(**a**) (**b**)

Figure 6. Comparison of experimental results for COCO val and YOGA val datasets, both with an image input size of 256 × 192. (**a**) Comparison on COCO val; (**b**) comparison on YOGA val. On both datasets, the SEPAM_HRNet model shows excellent performance.

In Figure 6a, the vertical axis represents the accuracy rate, the horizontal axis represents the computation amount, and the circle size corresponds to the model's parameter amount. It can be observed that the SEPAM_HRNet model proposed in this paper outperformed the recent human pose-estimation model, in terms of accuracy.

4.2. Experimental Results of the YOGA2022 Dataset

4.2.1. YOGA2022 Dataset Description

In response to the national call for physical education and sports and the call to enrich the diversity of physical education and sports for students, yoga has been added to the curricula of the vast majority of schools. As a healthy method of physical and mental exercise, yoga is practiced by more and more students. However, during yoga teaching, it is important to ensure that students correctly master each yoga practice posture to avoid injuries to body muscles, which are caused by incorrect postures. However, manually managing and evaluating these postures requires significant human and material resources, and manual evaluation lacks objectivity, due to subjective factors. Therefore, accurately predicting the keypoints of the human body can help teaching staff to better assess the normality of students' yoga postures. It is particularly noteworthy that the complexity of yoga poses may lead to the hidden nature of the human body's keypoints, making the human pose-estimation model more challenging to predict.

In order to adapt to a universal yoga scenario, this study created an indoor human yoga pose-characterization dataset, YOGA2022, for modeling experiments. The dataset contains ten types of yoga poses: Warrior I, Warrior II, Bridge, Downward Dog, Flat, Inclined Plate, Seated, Triangle, Phantom Chair, and Goddess. Each pose was captured in an indoor environment, as shown in Figure 7. During data collection, the camera captured video clips of left and right direction movements and frontal movements performed from the side or front. The production of this dataset provides an important reference resource for the teaching of yoga and the development of models for human pose estimation. The data-acquisition steps were as follows:

1. Yoga movement video capture. We used a professional video camera to film participants of different heights, recording ten yoga poses as Warrior I Side, Warrior II Side, Bridge Side, Downward Dog Side, Flat Side, Inclined Plate Side, Seated Positive, Triangle Side, Phantom Chair Side, and Goddess Positive to video files. Each movement was recorded from either a frontal or lateral angle to capture full movement detail. Example images of the dataset are shown in Figure 7, and these images will form the basis for subsequent experiments.
2. Frame extraction. Image frames were extracted from the recorded video to construct the dataset. In order to maintain data diversity and reduce redundancy, one image frame was extracted from the video every six frames. Since the frame rate of the

video was about five frames per second, such an extraction ensured balanced data and coverage. Each image contained only one participant, representing a separate image of a yoga pose.

3. Image augmentation. Five ways of expanding the image data were used to increase the diversity and robustness of the dataset: rotation, flipping, noise, brightening, and darkening. By rotating the image, different angles of observation were simulated, thus increasing the diversity of the data. Flipping the image helped the model learn symmetry and inverse action. Introducing noise made the model more robust to disturbances. Brightening and darkening the image simulated the image under different lighting conditions, which in turn increased the generalization ability of the data. With these five data enhancement approaches, the dataset was successfully extended, and a total of 15,350 images were obtained. Such data enhancement strategies helped to improve the generalization performance of the model and made it better adapted to yoga-movement images from different scenes and postures.

Figure 7. Example diagram of YOGA2022 dataset. The example figure of the dataset was selected from the yoga poses demonstrated by the four participants.

Through the above steps, an image set containing ten kinds of yoga movement postures, each with a size of 960 × 544, was successfully captured. In this dataset, each movement has multiple angles and diversified image data, which provides a rich sample resource for the study of yoga-movement postures. Such a dataset can provide more comprehensive information for model training and evaluation, which helps to improve the performance and generalization ability of the yoga-movement pose-prediction model.

After the data acquisition was completed, the captured image data were processed using the COCO Annotator annotation tool in order to extract the keypoint features of the person. The specific steps of data annotation processing were as follows: first, the human body frame was manually annotated; then, the locations of all keypoints were annotated, along with setting the visibility of the keypoints. The labeled information included the relative coordinates and size of the human target frame (x-coordinate, y-coordinate, frame width, and frame height), as well as the relative coordinates and visibility of the 17 human keypoints (x, y, and visibility). The dataset will be publicly released at https://github.com/zhangdandan-git/YOGA_POSE (accessed on 30 July 2023). We hope this dataset will serve as a valuable resource for other researchers.

4.2.2. Evaluation Criteria and Training Details

The YOGA2022 dataset contains a total of more than 15,000 images, and each image contains one person instance; the dataset labeling is the same as the COCO dataset labeling, which also contains 17 keypoints, and the keypoint labeling is the same as that of COCO. The experiments in this paper trained the model on the YOGA train2022 dataset (total of 11,788 images) and evaluated the model on the val2022 set (total of 3562 images). The experiments used the OKS-based evaluation criteria, and the loss function was the same as that in the COCO experiment. The details of the experiments on the YOGA2022 dataset were the same as those of the COCO dataset experiments, using the same parameter configurations and experimental environment.

4.2.3. Experimental Verification and Analysis

The experimental results on the YOGA2022 calibration set are displayed in Table 2. The results show that the SEPAM_HRNet model achieved superior results with lower computational complexity, compared to the recent human pose-estimation models PRTR, DistilPose-L, HRNet, and simple baseline. On mAP, the SEPAM_HRNet model outperformed the HRNet model by 0.2 percentage points, while achieving the best results on the other validation criteria. It is worth noting that the images in the YOGA dataset contained only one person instance and were large-sized targets in the images, so there were no prediction results for medium-sized targets in the experimental results for the YOGA2022 dataset. These experimental results further demonstrated the superiority of the SEPAM_HRNet model for the task of yoga-movement pose estimation and validated the specificity and uniqueness of the constructed dataset. A comparison of the experimental results is shown in Figure 6b.

Table 2. Performance comparison on YOGA2022 val2022. Comparison between SEPAM_HRNet and other methods on the YOGA2022 validation set.

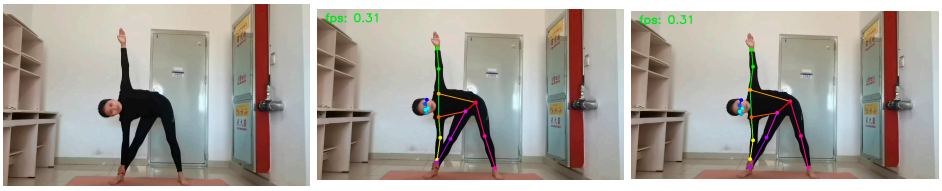
Models	Backbone	Input_Size	Params	GFLOPS	AP	AP.5	AP.75	AP(M)	AP(L)
Simple baseline	ResNet-50	256 × 192	34 M	8.9 G	91.3	92.9	91.9	−1.000	91.3
PRTR	HRNet	256 × 192	57.2 M	10.2 G	89.6	92.8	90.7	−1.000	89.6
DistilPose-L		256 × 192	**21.3 M**	10.3 G	86.3	90.9	88.7	−1.000	86.3
HRNet	HRNet	256 × 192	28.5 M	7.1 G	91.2	93.0	91.0	−1.000	91.2
SEPAM_HRNet		256 × 192	28.8 M	**7.0 G**	**91.4**	**93.0**	**92.0**	−1.000	**91.4**

5. Visualization Research and Analysis

In this paper, a visualization study was conducted on the YOGA2022 calibration set to demonstrate the effectiveness of prediction of occluded human keypoints. Ten images were randomly selected from the YOGA2022 dataset, each representing a yoga pose. The visualization results are shown in Figures 8–17, where the dots indicate the locations of the keypoints of the human body, and the connecting lines indicate the modeling of the keypoint relationships. These visualization results intuitively demonstrate the performance of the model in yoga-pose estimation.

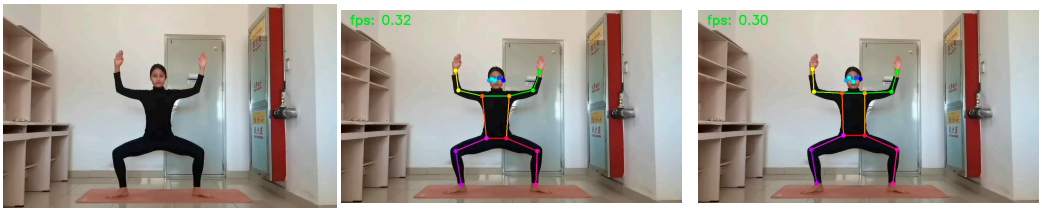
(**a**) (**b**) (**c**)

Figure 8. Warrior I: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. For the prediction of unobscured limb keypoints, the predictions of the two models were similar.



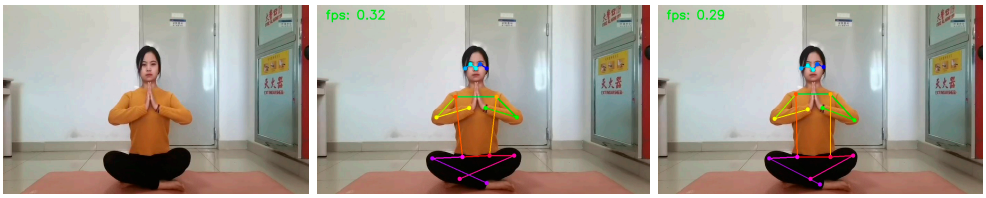
(**a**) (**b**) (**c**)

Figure 9. Triangle: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. For the prediction of unobscured limb keypoints, the predictions of the two models were similar.



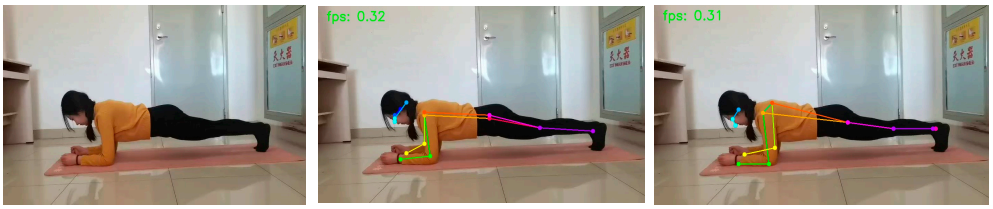
(**a**) (**b**) (**c**)

Figure 10. Goddess: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. For the prediction of unobscured limb keypoints, the predictions of the two models were similar.



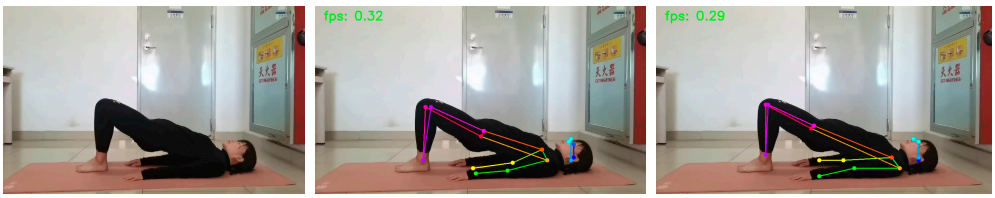
(**a**) (**b**) (**c**)

Figure 11. Seated: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performed more accurately in predicting the occluded left ankle keypoint.



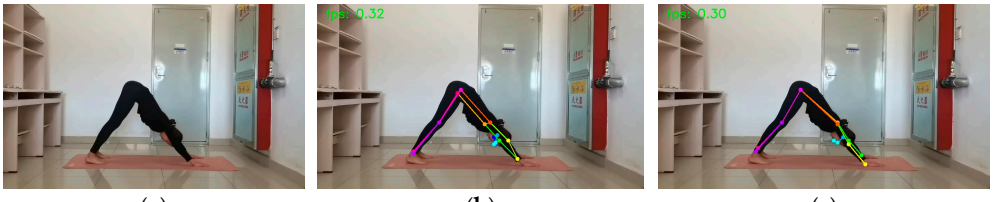
(**a**) (**b**) (**c**)

Figure 12. Flat: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performs more accurately in predicting occluded keypoints.



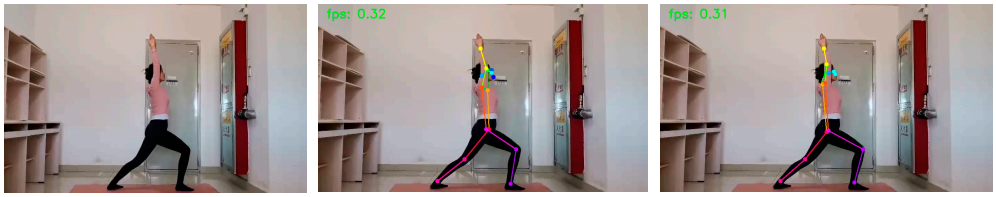
(**a**) (**b**) (**c**)

Figure 13. Bridge: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performed more accurately in predicting the occluded leg keypoints, but showed some bias in predicting the left elbow keypoints due to lighting and shadows.



(**a**) (**b**) (**c**)

Figure 14. Downward Dog: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performs more accurately in predicting occluded keypoints.

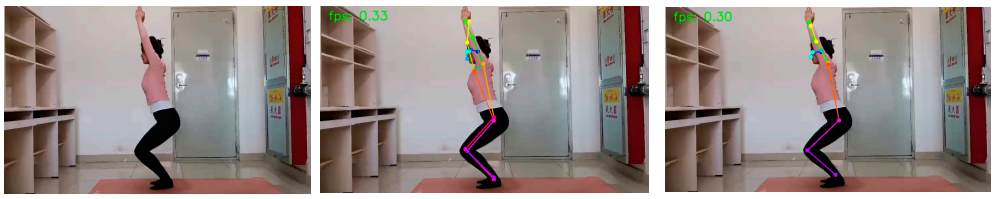


(**a**) (**b**) (**c**)

Figure 15. Warrior II: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performs more accurately in predicting occluded keypoints.

(**a**) (**b**) (**c**)

Figure 16. Inclined plate: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performs more accurately in predicting occluded keypoints.



(**a**) (**b**) (**c**)

Figure 17. Phantom Chair: (**a**) original picture; (**b**) HRNet prediction; (**c**) SEPAM_HRNet prediction. The SEPAM_HRNet model performs more accurately in predicting occluded keypoints.

Figures 8–10 show the frontal images of three yoga postures, Warrior One, Triangle Pose, and Goddess Pose, which are characterized by the absence of occluded limb keypoints. The prediction results show that both the SEPAM_HRNet model and the HRNet model can accurately predict the keypoints of the human body. The SEPAM_HRNet model is more detailed in predicting keypoints at the shoulders and thighs, while the two models have comparable performances in predicting the other keypoints.

Figures 11–17 show seven yoga postures that are characterized by the presence of occluded keypoints. The experimental results show that the SEPAM_HRNet model, after adopting the SEPAM module to improve the HRNet model, extracted multi-scale feature spatial information and more detailed feature information through the feature pyramid structure and further integrated the channel attention (SE) and the pixel attention mask (PAM). Such an improvement significantly enhances the extraction of the feature-map channel and the pixel-spatial-orientation information, improving the model's predictive performance at more minor scales and with more masks. Figure 11 shows the seated yoga pose in which the left ankle is occluded. Although both models predicted the left ankle, the SEPAM_HRNet model was more accurate. For the side yoga poses in Figures 12–17, almost half of the body's keypoints were in the occluded state. In these cases, the prediction results of the HRNet model deviated significantly from the correct human keypoint locations, but the SEPAM_HRNet model still managed to make correct predictions for these occluded shoulders, thighs, wrists, and elbows. The deviation of the SEPAM_HRNet model's prediction of the left elbow keypoint in Figure 13c was due to poor lighting conditions and shadows that resulted in unclear human features in the image, which affected the model's keypoint prediction accuracy. Overall, the SEPAM_HRNet model had better generalization ability and anti-interference ability than the HRNet model, as it could still predict keypoints and correctly model keypoint relationships when most human keypoints were

occluded. This indicated that the SEPAM_HRNet model has higher prediction accuracy and robustness when dealing with complex human postures and occluded situations.

6. Conclusions

In this paper, the SEPAMneck module and the SEPAMblock module were constructed by introducing the SEPAM module to improve the basic modules bottleneck and Basicblock in the HRNet model, thus proposing the SEPAM_HRNet human posture estimation network. This method effectively utilized the advantage of a feature pyramid to extract features while reducing the model arithmetic complexity, which ensured feature information extraction from the feature map. Meanwhile, the method utilized the channel attention and pixel-attention-mask map to extract better essential features in the channel dimension, which further improved the feature representation in the keypoint region and detailed the pixel location information. The experimental results validated the effectiveness of the SEPAM_HRNet model proposed in this paper, which outperformed other recent human pose-estimation models in predicting human keypoints with small scales and occlusions. In addition, we created the YOGA2022 dataset, which is specifically designed to study human yoga postures and enhance yoga teaching practices. This research is of great value to the yoga community and has the potential to benefit other applications that require accurate estimation of human poses under occlusion. In yoga, the SEPAM_HRNet model allows for a more accurate and detailed analysis of human yoga postures, which could help enhance yoga teaching practices by providing better instruction and feedback, improving practitioner performance, and reducing the risk of injury. The ability to accurately estimate human postures in the presence of occlusions has broader implications beyond yoga. In motion analysis, fitness tracking, and rehabilitation, the SEPAM_HRNet model can track human movement more accurately and reliably, helping coaches, athletes, and healthcare professionals analyze performance, monitor progress, and design personalized training or rehabilitation programs. The YOGA2022 dataset, which was explicitly created for studying human yoga postures, further contributes to this field of research by providing a standardized benchmark for evaluating the accuracy of models for estimating human postures in yoga, facilitating comparisons between different models and techniques.

Designing lightweight human pose-estimation models that are more applicable to real-world scenarios, while ensuring the accuracy of human keypoints prediction, is a future research direction. Lightweight models are more efficient and practical in real-world applications, so how to strike a balance between prediction accuracy and model complexity and how to design high-performance, lightweight human pose-estimation models suitable for real-world scenarios will be the foci of future research. In conclusion, this study advances the field of human pose estimation and has practical implications for the yoga community and other fields that rely on accurate and robust human pose estimation in occluded situations.

Author Contributions: Writing—review & editing, J.L., D.Z., L.S. and T.K.; Supervision, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Dataset address: https://github.com/zhangdandan-git/YOGA_POSE.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Toshev, A.; Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1653–1660.
2. Newell, A.; Yang, K.Y.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the 2016 European Conference on Computer Vision, LNCS 9912, Amsterdam, The Netherlands, 10–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 483–499.
3. Xiao, B.; Wu, H.P.; Wei, Y.C. Simple baselines for human pose estimation and tracking. In Proceedings of the 2018 European Conference on Computer Vision, LNCS 11210, Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 472–487.
4. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 5686–5696.
5. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
6. Woo, S.; Park, J.; Lee, J.; Kweon, I. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020.
7. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27 October–2 November 2019.
8. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
9. Andriluka, M.; Roth, S.; Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1014–1021.
10. Rothrock, B.; Park, S.; Zhu, S.C. In-tegrating grammar and segmentation for human pose estima-tion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–27 June 2013; pp. 3214–3221.
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
12. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
13. Fang, H.S.; Xie, S.; Tai, Y.W.; Lu, C. RMPE: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2353–2362.
14. Papandreou, G.; Zhu, T.; Chen, L.C.; Gidaris, S.; Tompson, J.; Murphy, K. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 282–299.
15. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
16. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 14–19 June 2020; pp. 5386–5395.
17. Xue, N.; Wu, T.; Xia, G.; Zhang, L. Learning Local-Global Contextual Adaptation for Multi-Person Pose Estimation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 13055–13064.
18. Maji, D.; Nagori, S.; Mathew, M.; Poddar, D. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Online, 14–19 June 2020; pp. 2636–2645.
19. Li, Y.; Zhang, S.; Wang, Z.; Yang, S.; Yang, W.; Xia, S.T.; Zhou, E. Tokenpose: Learning keypoint tokens for human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 10–17 October 2021; pp. 11313–11322.
20. Yang, S.; Quan, Z.; Nie, M.; Yang, W. Trans-pose: Keypoint localization via transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 10–17 October 2021; pp. 11802–11812.
21. Xu, Y.; Zhang, J.; Zhang, Q.; Tao, D. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv* **2022**, arXiv:2204.12484.
22. Feng, R.; Gao, Y.; Ma, X.; Tse, T.H.; Chang, H.J. Mutual Information-Based Temporal Difference Learning for Human Pose Estimation in Video. *arXiv* **2023**, arXiv:2303.08475.
23. He, Q.; Yang, L.; Gu, K.; Lin, Q.; Yao, A. Analyzing and Diagnosing Pose Estimation with Attributions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 4821–4830.

24. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-hrnet: A lightweight high-resolution network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 10440–10450.
25. Huang, J.; Zhu, Z.; Guo, F.; Huang, G. The devil is in the details: Delving into unbiased data processing for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 14–19 June 2020; pp. 5700–5709.
26. Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; Zhu, C. Distribution-aware coordinate representation for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 14–19 June 2020; pp. 7093–7102.
27. Neff, C.; Sheth, A.; Furgurson, S.; Tabkhi, H. Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation. *arXiv* **2020**, arXiv:2007.08090.
28. Wang, Y.; Li, M.; Cai, H.; Chen, W.; Han, S. Lite Pose: Efficient Architecture Design for 2D Human Pose Estimation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 13116–13126.
29. Nie, X.; Feng, J.; Zhang, J.; Yan, S. Single-stage multi-person pose machines. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
30. Li, Z.; Ye, J.; Song, M.; Huang, Y.; Pan, Z. Online Knowledge Distillation for Efficient Pose Estimation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Online, 10–17 October 2021; pp. 11720–11730.
31. Hong, J.; Fisher, M.; Gharbi, M.; Fatahalian, K. Video Pose Distillation for Few-Shot, Fine-Grained Sports Action Recognition. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Online, 10–17 October 2021; pp. 9234–9243.
32. Ye, S.; Zhang, Y.; Hu, J.; Cao, L.; Zhang, S.; Shen, L.; Wang, J.; Ding, S.; Ji, R. DistilPose: Tokenized Pose Regression with Heatmap Distillation. *arXiv* **2023**, arXiv:2303.02455.
33. Mao, W.; Ge, Y.; Shen, C.; Tian, Z.; Wang, X.; Wang, Z. Tfpose: Direct human pose estimation with transformers. *arXiv* **2021**, arXiv:2103.15320.
34. Mao, W.; Ge, Y.; Shen, C.; Tian, Z.; Wang, X.; Wang, Z.; den Hengel, A.V. Poseur: Direct human pose regression with transformers. *arXiv* **2022**, arXiv:2201.07412.
35. Li, K.; Wang, S.; Zhang, X.; Xu, Y.; Xu, W.; Tu, Z. Pose recognition with cascade transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 1944–1953.
36. Panteleris, P.; Argyros, A. Pe-former: Pose estimation transformer. In Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence, Paris, France, 1–3 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 3–14.
37. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective Kernel Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 510–519.
38. Zhang, H.; Zu, K.; Lu, J.; Zou, Y.; Meng, D. EPSANet: An efficient pyramid squeeze attention block on convolutional neural network. In Proceedings of the Asian Conference on Computer Vision (ACCV), Macau, China, 4–8 December 2022; pp. 1161–1177.
39. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded Pyramid Network for Multi-person Pose Estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7103–7112.