Lightweight Human Pose Estimation Based on Heatmap Weighted Loss Function

Xin Wang, Guanhua Li, Yongfeng Chen, Ge Wen

***Abstract*—Current research on human pose estimation often focuses on using complex structures to improve task accuracy, while overlooking resource consumption and inference speed during actual deployment. Based on the LitePose pose estimation architecture, this paper proposes a lightweight bottom-up pose estimation model, WLitePose, designed to better handle complex scenes. Specifically, to address the limitations of the MSE loss function, a heatmap weighted loss function is proposed to enable the model to focus more on the areas surrounding the true keypoint locations during training. To enhance the model's ability to handle variations in human scale, a lightweight deconvolution module is used after the main architecture to generate higher-resolution heatmaps. During the inference phase, heatmaps of different resolutions are aggregated. Additionally, the DFC-bottleneck block is proposed to enhance the backbone network's ability to capture long-range dependence between different spatial pixels. Experimental results on the COCO and CrowdPose datasets demonstrate that the proposed model achieves a good balance between task accuracy and computational complexity.**

***Index Terms*—bottom-up human pose estimation, lightweight network, multi-resolution heatmap aggregation, long-range dependence**

I. Introduction

HUMAN pose estimation aims to perform keypoint localization on individuals in images. This technology has a wide range of applications, such as action recognition [1], abnormal behavior detection [2], [3], [4], and medical assistance [5], [6]. Therefore, improving the accuracy of this technology is highly beneficial for other downstream tasks. Human pose estimation models generally fall into two primary categories: top-down approaches and bottom-up methods. Top-down frameworks [7], [8], [9], [10] first detect individual persons and then perform keypoint localization for each person. Conversely, the bottom-up framework [11], [12], [13], [14] initially predicts keypoints without identity information, which are then associated with the corresponding individuals. In complex and crowded scenarios, the bottom-up approach is faster and more robust.

Multi-scale processing and high-resolution features help retain more positional information, which is highly beneficial for keypoint localization. This has also been a key focus of many recent works. The Hourglass network [15] enhanced accuracy by stacking multiple hourglass modules using an intermediate supervision strategy. HRNet [9] connects multi-resolution subnetworks in parallel, retaining feature information at multiple resolution levels to generate high-quality heatmaps. HigherHRNet [13] built upon HRNet to further enhance task accuracy through heatmap aggregation. Furthermore, there are many other methods based on HRNet. SWAHR [16] started from the perspective of adjusting the standard deviation of the Gaussian kernel, while DEKR [17] posited that accurately regressing keypoint positions required focusing on the areas surrounding the keypoints. These methods have achieved good results in improving task accuracy from different angles. However, all the aforementioned methods have high computational complexity, which affects the speed of model training and inference.

Currently, mainstream research primarily focuses on model accuracy while neglecting real-time performance, which makes it difficult to run on edge devices. This limitation restricts the application scenarios of the models to some extent. Therefore, designing lightweight human pose estimation methods is also important. Lightweight OpenPose [18] modified the two-branch structure of OpenPose and used both a lightweight backbone network and small convolutional kernels, significantly reducing computational effort. ViPNAS [19] advanced the development of lightweight models by leveraging Neural Architecture Search (NAS) techniques.

HRNet has become a benchmark model for many works in large human pose estimation networks, owing to the advantages of its unique structure. It is equally important in lightweight networks. EfficientHRNet [20] applied the idea of EfficientNet [21] to HRNet to achieve a lightweight treatment of HRNet. Lite-HRNet [22] improved HRNet by using enhanced shuffle blocks, significantly reducing computational costs. Dite-HRNet [23] introduced a dynamic inference mechanism, making HRNet more efficient during inference, thereby adapting to a wider range of practical application scenarios. The aforementioned methods are theoretically lightweight, but their performance on edge devices is suboptimal due to the parallel multi-branch structures. Additionally, computing weights between different resolutions can further reduce the model's inference speed. LitePose [24] validated through gradual shrinkage

Manuscript received April 24, 2024; revised September 30, 2024.

This work was supported in part by the Basic Research Key Project of Liaoning Provincial Department of Education (LJ212410153035) and the Shenyang Science and Technology Planning Project (22-322-3-36).

Xin Wang is an associate professor of the School of Electrical and Control Engineering, Shenyang Jianzhu University, Shenyang, China. (e-mail: wangx7988@sjzu.edu.cn).

Guanhua Li is a postgraduate student of the School of Electrical and Control Engineering, Shenyang Jianzhu University, Shenyang, China. (e-mail: 2495823000@qq.com).

Yongfeng Chen is a professor of the Internet Business Department, Hebei Software Institute, Baoding, China. (corresponding author to provide e-mail: 13513420447@163.com).

Ge Wen is a senior engineer of the Information and Communication Engineering Design Institute, Shenyang, China. (e-mail: 2026678321@qq.com).

experiments that parallel multi-branch structures were not suitable for edge devices. It designed an efficient single-branch architecture to avoid the redundant refinement in the fusion modules of multi-branch architectures. Therefore, it is evident that research on lightweight human pose estimation remains insufficient compared to mainstream approaches.

This paper proposes WLitePose, a lightweight model for human pose estimation based on the LitePose architecture, aimed at better handling complex and crowded environments. Specifically, different pixels in the heatmap have varying importance for keypoint localization, whereas the MSE loss function assigns equal importance to each pixel, ignoring these differences. To address this, a heatmap weighted loss function is proposed to differentiate the varying importance of different pixels, enabling the model to focus more on the pixels surrounding the true keypoint locations. For smaller-scale individuals, higher-resolution heatmaps contain more detailed keypoint location information. Therefore, a lightweight deconvolution module is designed after the main architecture to generate higher-resolution heatmaps. Although the design of depthwise separable convolutions in the backbone network reduces computational complexity, it also limits the network's ability to capture extensive and complex spatial context information. To address this, the basic block of the backbone network has been redesigned, proposing the DFC-bottleneck block to enhance the network's ability to capture long-range dependence between different spatial pixels. These improvements allow the proposed model to achieve superior performance on the public datasets. The main contributions of this paper are summarized as follows:

- This paper introduces WLitePose, a new lightweight human pose estimation model that utilizes a heatmap weighted loss function to assign varying weights to pixels across the heatmap. This allows the model to concentrate more on the areas around the true keypoint locations during training.
- A lightweight deconvolution module is employed after the main architecture to generate higher-resolution heatmaps. During the inference phase, heatmaps with 1/4 and 1/2 resolutions are aggregated to better predict the keypoints of smaller-scale individuals.
- The DFC-bottleneck block is proposed to enhance the feature extraction capability of the backbone network, thereby improving the overall accuracy of the model.
- The experimental results show that the task accuracy of the proposed model outperforms other lightweight models.

II. Related Work

A. Bottom-up Human Pose Estimation

Given the advantages of the bottom-up approach in multi-user scenarios, this paradigm is also more suitable for running on edge devices. Currently, the mainstream research direction still focuses primarily on improving accuracy. Du et al. [25] proposed a new framework from the perspective of heatmap encoding and decoding. Jin et al. [26] classified keypoints as belonging to a single individual when their offsets from the body center exhibit consistency. Yu et al. [27] improved task accuracy by adjusting the response area of Gaussian distributions. While the aforementioned methods demonstrate commendable accuracy, deploying them on edge devices proves challenging. Currently, there is a noticeable lack of research on lightweight network models.

B. Design of Lightweight Networks

In industrial research, designing lightweight deep neural network architectures has been a prevalent focus. Building on the ability of depthwise separable convolutions to significantly reduce parameters and computational load, MobileNet [28] made its debut. Following MobileNet, MobileNetV2 [29] introduced inverted residual blocks, while MobileNetV3 [30] leveraged neural architecture search (NAS) techniques to further refine the network structure. ShuffleNet [31] introduced channel shuffle operations to facilitate information exchange between feature map groups. GhostNetV2 [32] integrated self-attention mechanisms into Ghost modules to capture dependence between distant pixels. The lightweight networks mentioned above have been widely applied in visual tasks, providing significant inspiration for the work presented in this paper.

C. Attention Mechanism

Attention mechanisms have garnered significant interest in research in recent years. The SE channel attention mechanism module proposed by Hu et al. [33] enhances the network's representational capacity by dynamically adjusting the weights of each channel. Woo et al. [34] introduced the CBAM module, which adaptively adjusts the importance of channel and spatial dimensions in feature maps. The CA module [35] has further promoted the advancement of attention mechanisms. The attention mechanism is also very helpful for pose estimation. Xu et al. [36] achieved the current highest accuracy by integrating the vision transformer architecture into the task.

III. Methodology

Figure 1 illustrates the network structure of the proposed lightweight model, WLitePose. The backbone network adopts the MobileNetV2 architecture, with the basic block replaced by the proposed DFC-bottleneck block. Through stages 1 to 4, feature maps with a resolution of 1/32 of the original input image size are obtained. Subsequently, multiple scales of feature fusion and deconvolution are performed to obtain feature maps with a resolution of 1/4 of the input image size. These feature maps are then input into the lightweight deconvolution module designed in this study to generate feature maps with a resolution of 1/2 of the input image size. During the training phase, a multi-resolution supervision strategy is employed, calculating the loss between ground truth heatmaps and predicted heatmaps separately at these two resolutions. During the inference phase, the predicted heatmaps at 1/4 and 1/2 resolutions are aggregated and averaged to obtain the final heatmap. Keypoint position information is then obtained via heatmap decoding. Keypoint grouping is done using the Associative

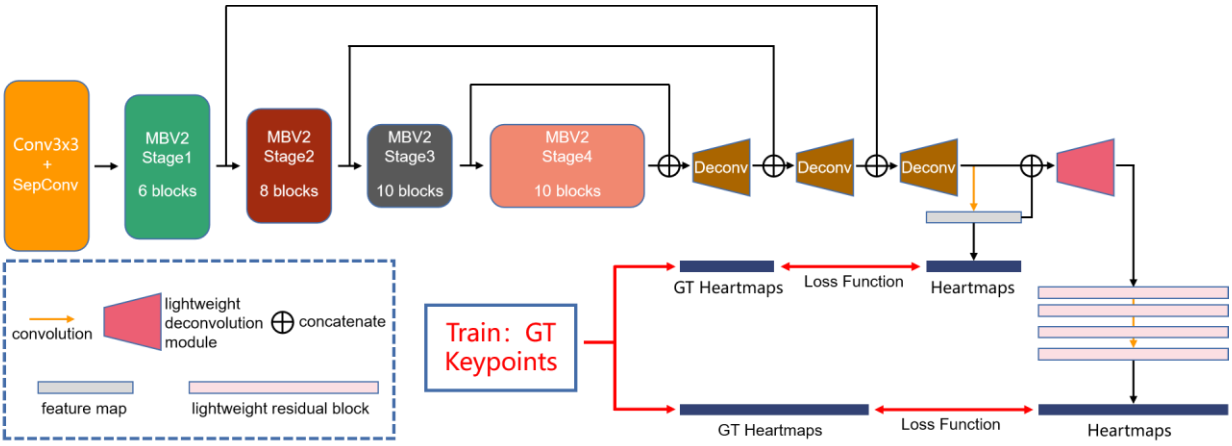

Figure 1. The architecture of WLitePose

Embedding [11] method.

A. *Heatmap Weighted Loss Function*

Since the MSE loss function is continuous and differentiable, it is very suitable for supervising the training of neural networks. Therefore, in the field of human pose estimation, the most commonly used loss function in the training phase of network models is the MSE function, which is expressed as follows:

$$L_{\text{regression}} = \|P - H\|_2^2 \tag{1}$$

where P is the value of the predicted heatmap and H is the value of the ground truth heatmap. Human pose estimation models based on heatmap regression generate ground truth heatmaps using Gaussian functions to cover the surrounding areas of each keypoint. The closer a pixel is to the actual keypoint location, the closer its corresponding ground truth heatmap value is to 1. Using the traditional MSE loss function, equal weights are assigned to each pixel in the heatmap, ignoring the varying contributions of different pixels to the keypoint localization task. This paper argues that pixels in different locations on the heatmap have varying levels of importance for keypoint localization. The weights of pixels at different positions on the ground truth heatmap should be distinguished, with the model paying more attention to pixels with higher values on the ground truth heatmap, which are closer to the actual keypoint positions. Inspired by Focal Loss [37], this study proposes a heatmap weighted loss function by adding a weighting coefficient to the MSE loss function as follows:

$$L_{\text{regression}} = W \times \|P - H\|_2^2 \tag{2}$$

The weight W is defined as follows:

$$W = \begin{cases} 1 - P_{k,i,j} & \{k, i, j\} \text{ is positive sample} \\ P_{k,i,j} & \{k, i, j\} \text{ is negative sample} \end{cases} \tag{3}$$

where $\{k, i, j\}$ represent the pixel positions in the predicted heatmap P. The heatmap values are continuous, which makes it difficult to define the boundaries between positive and negative samples. Therefore, the weight W is defined as follows:

$$W = \|P\| \cdot \left(1 - (H)^{\tau}\right) + \|1 - P\| \cdot (H)^{\tau} \tag{4}$$

$\|\cdot\|$ is an absolute value function. Define τ as a hyperparameter governing the position of the soft boundary, with the ground truth heatmap value θ serving as the threshold for distinguishing the soft boundaries between positive and negative samples, where $1 - \theta^{\tau} = \theta^{\tau}$. In the ground truth heatmap, the closer a pixel is to the actual keypoint location, the higher its corresponding heatmap value. For samples with ground truth heatmap values greater than the threshold θ, the $(H)^{\tau}$ value approaches 1, and its loss weight is closer to (1−P). This indicates that the pixels in this region are positive samples, and the closer they are to the actual keypoint position, the more the model will focus on those pixels. For samples with ground truth heatmap values less than the threshold θ, the loss weight is closer to P. This indicates that the pixels in this region are negative samples, and the model's attention to these pixels will decrease. By assigning different weights to pixels in various locations on the heatmap, the model focuses more on the regions closer to the actual keypoint locations, thereby enhancing keypoint localization accuracy. In the experiment, the value of τ was taken as 0.01.

B. *Lightweight Deconvolution Module*

To enhance the model's ability to handle variations in human scale and better predict keypoints for smaller-scale individuals, this paper, inspired by HigherHRNet [13], designs a deconvolution module after the LitePose [24] architecture to generate higher-resolution heatmaps. Without changing the Gaussian kernel standard deviation and without affecting the keypoint localization accuracy of larger individuals, this approach improves the localization accuracy of keypoints for medium and small-scale individuals, thereby enhancing the overall performance of the model. The final feature map generated by the main architecture has a resolution of 1/4 of the input image, which is used as the input for the added deconvolution module to generate a new feature map at half the original image's resolution. These two high-resolution feature maps at different scales form a feature pyramid. For smaller-scale individuals, the 1/8 resolution heatmap contains less keypoint position information compared to the 1/2 resolution heatmap. Therefore, during the inference phase, the proposed model's aggregation of the 1/4 and 1/2 resolution heatmaps performs better than the original framework's aggregation of the 1/8 and 1/4 resolution heatmaps.

In the deconvolution module, a 4×4 convolutional kernel is used for the transposed convolution, followed by batch normalization (BN) and ReLU activation applied to the

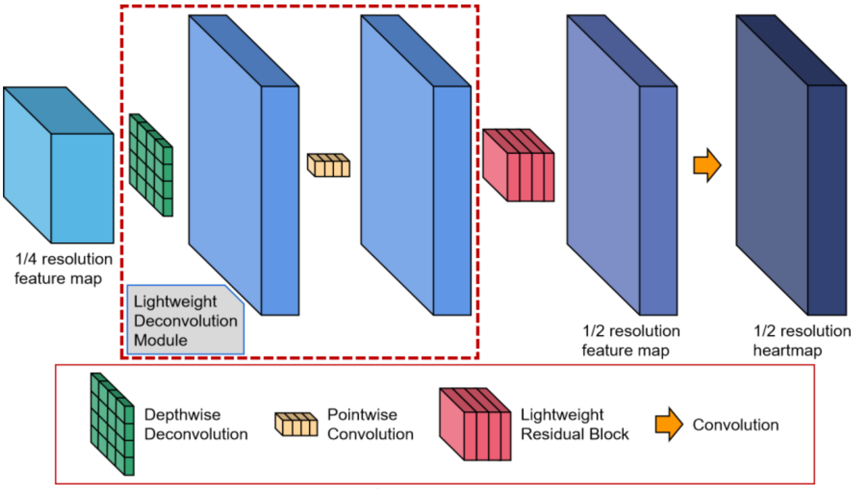
1/4 resolution feature map
Lightweight Deconvolution Module
1/2 resolution feature map
1/2 resolution heartmap
Depthwise Deconvolution
Pointwise Convolution
Lightweight Residual Block
Convolution

Figure 2. The structure of the lightweight deconvolution module

upsampled feature maps. To obtain high-quality feature maps, four residual blocks are added after the deconvolution to further refine the features. Considering the computational cost, the deconvolution module and the residual blocks are designed to be lightweight. The parameters of a standard deconvolutional layer are defined by the convolutional kernel K's size, represented as $D_K \times D_K \times C_{in} \times C_{out}$. Its computational cost depends mainly on the size of the convolutional kernel $D_K \times D_K$, the number of input channels C_{in}, the number of output channels C_{out}, and the size W × H of the feature map. The computational cost of the *l*-layer deconvolutional layer can be expressed as:

$$\sum_{i}^{l} D_{K_i} \times D_{K_i} \times C_{in_i} \times C_{out_i} \times W_i \times H_i \qquad (5)$$

Inspired by MobileNet [28], this paper applies the idea of depthwise separable convolution to the deconvolution module. The standard deconvolutional layer is replaced with a depthwise deconvolutional layer, followed by a pointwise convolutional layer, as shown in Figure 2. The depthwise deconvolutional layer applies a convolutional kernel to each channel for upsampling, followed by a pointwise convolutional layer for cross-channel feature fusion. The computational volume of the lightweight deconvolution module is expressed as:

$$\sum_{i}^{l} D_{K_i} \times D_{K_i} \times C_{in_i} \times W_i \times H_i + C_{in_i} \times C_{out_i} \times W_i \times H_i \qquad (6)$$

In the added deconvolution module of this paper, there is only one deconvolutional layer, so $l = 1$. The convolution kernel has a size of 4×4, with 66 input channels and 32 output channels. Therefore, the ratio of computational volume between the lightweight deconvolution and the original deconvolution is approximately:

$$\frac{4\times4\times66\times W_i\times H_i+66\times32\times W_i\times H_i}{4\times4\times66\times32\times W_i\times H_i}=\frac{1}{32}+\frac{1}{16}=\frac{3}{32} \qquad (7)$$

Figure 3 illustrates the structure of the lightweight residual block. Inspired by ShuffleNet [31], the 1×1 convolution in the residual block can be replaced with 1×1 pointwise group convolution (GConv). By dividing the channels into 3 groups, the computational cost of pointwise convolutions after grouping is approximately 1/3 of the original. After pointwise group convolutions, the output result is subjected to channel shuffle, where subgroups between different color channels are rearranged into a new grouping. The following 3×3 convolution can be substituted with a 3×3 depthwise convolution (DWConv).

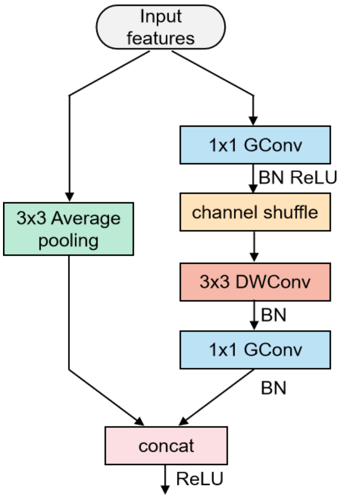

Figure 3. The structure of the lightweight residual block

C. DFC-bottleneck Block

In the inverted residual module of the backbone network MobileNetV2, LitePose uses 7×7 convolutional kernels to expand the receptive field, achieving notable results. In the inverted residual module, the feature channel dimensions are expanded using 1×1 pointwise convolution. However, during this process, the 1×1 convolutional layer only considers the information at each position in the feature map independently, lacking interaction with other pixels. Although the 7×7 depthwise convolution can expand the receptive field within a single channel, it operates solely within the respective channel, capturing only local spatial features within each channel. The absence of cross-channel interactions limits the network 's ability to capture complex spatial contextual information, preventing direct access to global information. This is detrimental to keypoint localization tasks in complex scenes or for small-scale targets. This paper introduces a novel bottleneck module, the DFC-bottleneck block, designed to improve the network's capacity to capture long-range dependencies between spatial pixels, thereby enhancing overall model accuracy.

The DFC attention was proposed by GhostNetV2 [32]. Given an input feature $Z \in R^{H\times W\times C}$, treat it as *HW* tokens, i.e.

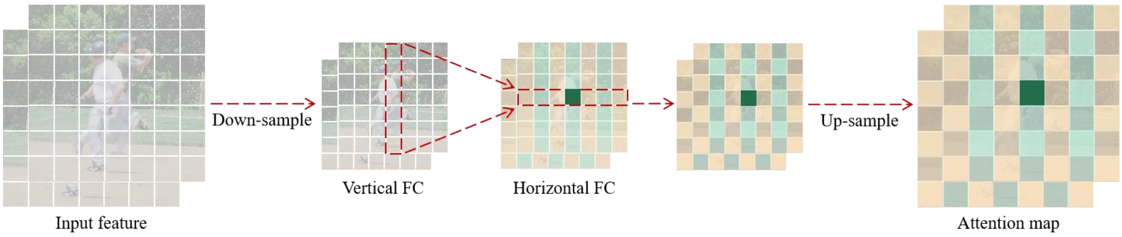

Figure 4. The structure of DFC attention

$z_i \in R^C$, $Z = \{z_{11}, z_{12}, \cdots, z_{HW}\}$. The formula for generating the attention map using a fully connected (FC) layer is:

$$a_{hw} = \sum_{h',w'} F_{hw,h'w'} \odot z_{h'w'} \tag{8}$$

where the symbol Θ denotes element-wise multiplication, F represents the learnable weights in the fully connected (FC) layer, and $A = \{a_{11}, a_{12}, \cdots, a_{HW}\}$ denotes the generated attention map. During the computation of attention output a_{hw} for each position, information from all other positions is integrated. Therefore, the equation above aggregates all tokens and learnable weights together to capture global information. Considering that feature maps in CNNs are typically low-rank, it is unnecessary to densely connect all input and output tokens from different spatial positions. Eq. (8) can be decomposed into two FC layers, denoted as:

$$a'_{hw} = \sum_{h'=1}^{H} F^{H}_{h,h'w} \odot z_{h'w},\ h = 1, 2, \cdots, H,\ w = 1,\ 2, \cdots,\ W \tag{9}$$

$$a_{hw} = \sum_{w'=1}^{W} F^{W}_{w,hw'} \odot a'_{hw'},\ h = 1, 2, \cdots, H,\ w = 1,\ 2, \cdots,\ W \tag{10}$$

where F^H and $F^W$are the transformation weights. With the original feature Z as input, Eq. (9) and Eq. (10) capture long-range dependence along the horizontal and vertical directions, respectively.

In this paper, the 7×7 convolutional kernel is still used. However, to better balance the relationship between accuracy and computation, and inspired by ConvNeXt [39], the 7×7 depthwise convolutional layer is moved before the 1×1 convolutional layer. While this approach may not achieve optimal accuracy, it appropriately reduces computational cost. The DFC attention mechanism takes a parallel structure with the 7×7 depthwise convolutional layer. The input feature $X \in R^{H\times W\times C}$ is fed into two separate branches. One branch is the 7×7 depthwise convolutional layer, and the other branch is the DFC module that generates the attention map A. The input feature X is transformed into the DFC's input Z using a 1×1 convolution. The final output $O \in R^{H\times W\times C}$ is the product of the outputs from the two branches, as follows:

$$O = \mathrm{Sigmoid}(A) \odot v(X) \tag{11}$$

where ⊙ denotes element-wise multiplication, and the purpose of the Sigmoid function is to normalize the output of the attention map A to the range of (0,1). Figure 4 illustrates the structure of DFC attention. To lower the computational cost of the parallel structure, the input features of the DFC branch are downsampled via average pooling, reducing the original width and height by half. Subsequent vertical FC and horizontal FC operations are then performed on these downsampled features. The resulting feature maps are upsampled to their original size to align with the resolution of the features from the other branch. Bilinear interpolation is employed for upsampling.

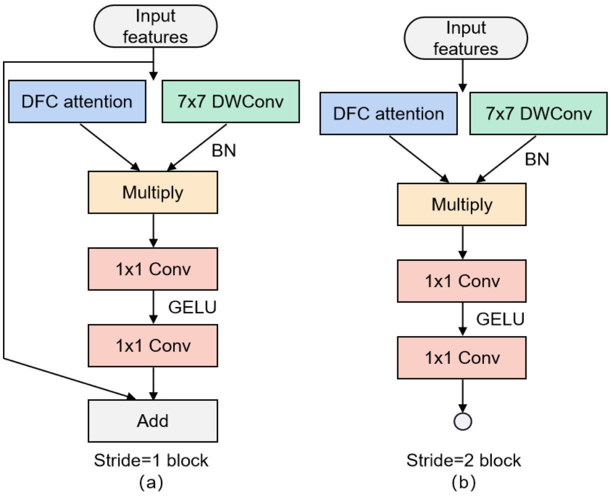

Figure 5. The structure of the DFC-bottleneck block

The structure of the DFC-bottleneck block is shown in Figure 5. In the DFC-bottleneck block, the DFC attention branch runs in parallel with the 7×7 depthwise convolutional layer. Subsequently, this paper is inspired by ConvNeXt [39] to fine-tune the module. The activation function is changed from ReLU to GELU, retaining only one GELU layer between the two 1×1 convolutional layers. Additionally, two BN layers are removed, leaving only one BN layer in front of the 1×1 convolutional layer. The accuracy loss caused by moving the 7×7 depthwise convolutional layer forward is appropriately compensated by these adjustment strategies above.

IV. Experiments

A. Datasets

1) Microsoft COCO: The MS-COCO dataset [38] consists of over 200K images, featuring 250K human body instances annotated with 17 keypoints. The training set consists of 57K images, with the validation and test sets containing 5K and 20K images, respectively.

2) CrowdPose: The CrowdPose dataset [8] contains 20K images. The human body instance is annotated with 14 keypoints. The ratio of the training, validation, and test sets is 5:1:4. The detection difficulty of this dataset is greater. Utilizing the approach of HigherHRNet [13], the training occurs using both the training and validation datasets, with subsequent evaluation conducted on the testing dataset.

B. Evaluation Metrics

This paper adopts the Object Keypoint Similarity (OKS) provided by the Microsoft COCO dataset [38] as the

TABLE I
COMPARISON OF EXPERIMENTAL RESULTS ON THE COCO VAL2017 SET

Bottom-up method	Backbone	Input Size	#Params/M	#MACs	Latency (ms)	AP	AP^{50}	AP^{75}
Large Networks								
PersonLab [12]	ResNet-152	1401×1401	68.7	405.5G	–	66.5	86.2	71.9
HigherHRNet [13]	HRNet-W32	512×512	28.6	47.9G	–	67.1	86.2	73.0
	HRNet-W48	640×640	63.8	155.1G	–	69.9	87.2	76.1
Small Networks								
Lightweight OpenPose [18]	–	368×368	4.1	9.0G	97	42.8	–	–
EfficientHRNet [20]	EfficientHRNet-H$_{-2}$	448×448	8.3	7.9G	182	52.8	72.2	57.9
	EfficientHRNet-H$_{-4}$	384×384	2.8	2.2G	–	35.7	–	–
LitePose [24]	LitePose-S	448×448	2.7	5.0G	76	56.4	77.2	61.2
	LitePose-XS	256×256	1.7	1.2G	27	40.1	61.3	42.8
WLitePose	WLitePose-S	448×448	3.6	5.8G	81	**57.7**	**78.2**	**63.0**
	WLitePose-XS	256×256	2.3	1.5G	29	41.1	62.1	44.3

evaluation standard. The definition of OKS is as follows:

$$\mathrm{OKS}=\frac{\sum_{i}\exp\left(-d_i^2/2s^2k_i^2\right)\delta\left(v_i>0\right)}{\sum_{i}\delta\left(v_i>0\right)} \tag{12}$$

where d_i denotes the Euclidean distance between the predicted keypoint position and the ground truth position, v_i denotes the visibility of keypoint i's ground truth position (i.e., $v_i > 0$ means that the keypoint is visible, and $v_i \leq 0$ means that the position of the keypoint is unobservable), s is the scale factor of the target, and k_i is the constant used for each keypoint to control the attenuation. In addition, the standard Average Precision AP (i.e., the average precision of keypoint prediction at OKS = 0.50, 0.55, ..., 0.90, 0.95), AP^{50} (precision at OKS = 0.5) and AP^{75} (precision at OKS = 0.75) will be used in this paper as the standard metrics.

C. *Experiment Settings*

The server used for the experiments was running on Ubuntu 16.04 and was equipped with two NVIDIA RTX 3090 GPUs. The software environment included PyTorch 1.8.0, CUDA 11.1, etc.

Following the experimental setup of LitePose [24], the Adam optimizer was used for training. The batch size was set to 32. For the COCO dataset, training was carried out over 500 epochs. The starting learning rate was established at 10^{-3}, which was subsequently decreased to 10^{-4} after the 350$_{\text{th}}$ epoch and further to 10^{-5} by the 480$_{\text{th}}$ epoch. For the CrowdPose dataset, training was carried out over 200 epochs. The starting learning rate was established at 10^{-3}, which was subsequently decreased to 10^{-4} after the 50$_{\text{th}}$ epoch and further to 10^{-5} by the 180$_{\text{th}}$ epoch. Data augmentation techniques included random rotation ([-30°, 30°]), random scaling ([0.75, 1.5]), random translation ([-40, 40]), and random flipping. Additionally, the model’s latency was tested on the Qualcomm Snapdragon 855 GPU.

D. *Main Results Analysis*

Table I shows the results of the comparative experiments on the COCO val2017 set. While WLitePose has a slightly lower AP score than the larger HigherHRNet [13], it has significantly fewer parameters and lower MACs. Compared to the Lightweight OpenPose [18], WLitePose improves the AP score by 14.9%. With an input image size of 448×448, WLitePose achieves a 4.9% higher AP score than EfficientHRNet [20] and a 1.3% higher AP score than LitePose [24]. For an input size of 256×256, it also surpasses LitePose by 1.0% in AP score. When tested on a Qualcomm Snapdragon 855 GPU, WLitePose demonstrated lower latency than both Lightweight OpenPose and EfficientHRNet. In comparison to LitePose, it only adds 5ms of latency with a 448×448 input size while achieving greater accuracy. With an input size of 256×256, the latency remains nearly identical to LitePose.

Figure 6 visually shows the latency on a Qualcomm Snapdragon 855 GPU and the accuracy on the COCO val2017 set for various lightweight networks. LOpenPose denotes Lightweight OpenPose, and EHRNet-H_2 represents EfficientHRNet-H$_{-2}$. It is evident that WLitePose-S achieves the highest AP score. Additionally, its latency is only 5ms higher than LitePose-S and significantly lower than Lightweight OpenPose and EfficientHRNet-H$_{-2}$. This also validates the superiority of WLitePose.

Table II shows the results of the comparative experiments on the COCO test-dev2017 set. Compared to large networks, the computational complexity of the proposed model remains at a relatively low level. In addition, compared to the lightweight OpenPose, the AP score increased by 14.7%. With an input image size of 448×448, the proposed model achieves a 4.8% higher AP score than EfficientHRNet and a 1.2% higher score than LitePose. For an input size of 256×256, the AP score is 0.8% higher than that of LitePose.

Table III presents the results of the comparative experiments on the CrowdPose test set. While the proposed model is less accurate than larger networks, it has significantly fewer parameters and lower MACs. Compared to LitePose, it achieves a 1.6% higher AP score with an input size of 448×448 and a 1.2% higher score with an input size of 256×256. Among lightweight networks, WLitePose excels in both accuracy and computational efficiency due to its streamlined single-branch architecture. When deployed on a Qualcomm Snapdragon 855 GPU, the model's latency is similar to LitePose with a 256×256 input size, and increases by only 5ms with a 448×448 input size. Although WLitePose-S theoretically exhibits higher MACs compared to EfficientHRNet-H$_{-3}$, its actual deployment latency is 51ms lower. This further validates that parallel multi-branch architectures are not well-suited for edge devices.

Figure 7 visually represents the latency on a Qualcomm Snapdragon 855 GPU and the accuracy on the CrowdPose test set for various lightweight networks. HHRNet-W16 and EHRNet-H_3 denote HigherHRNet-W16 and EfficientHRNet-H$_{-3}$, respectively. It is evident that

TABLE II
COMPARISON OF EXPERIMENTAL RESULTS ON THE COCO TEST-DEV2017 SET

Bottom-up method	Backbone	Input Size	#Params/M	#MACs	Latency (ms)	AP	AP^{50}	AP^{75}
Large Networks								
PersonLab [12]	ResNet-152	1401×1401	68.7	405.5G	–	66.5	88.0	72.6
HigherHRNet [13]	HRNet-W32	512×512	28.6	47.9G	–	66.4	87.5	72.8
	HRNet-W48	640×640	63.8	155.1G	–	68.4	88.2	75.1
Small Networks								
Lightweight OpenPose [18]	–	368×368	4.1	9.0G	97	42.7	–	–
EfficientHRNet [20]	EfficientHRNet-H$_{-2}$	448×448	8.3	7.9G	182	52.6	72.8	57.5
	EfficientHRNet-H$_{-4}$	384×384	2.8	2.2G	–	35.5	–	–
LitePose [24]	LitePose-S	448×448	2.7	5.0G	76	56.2	78.4	60.5
	LitePose-XS	256×256	1.7	1.2G	27	37.6	62.1	40.9
WLitePose	WLitePose-S	448×448	3.6	5.8G	81	**57.4**	**79.1**	**62.4**
	WLitePose-XS	256×256	2.3	1.5G	29	38.4	62.7	42.0

TABLE III
COMPARISON OF EXPERIMENTAL RESULTS ON THE CROWDPOSE TEST SET

Bottom-up method	Backbone	Input Size	#Params/M	#MACs	Latency (ms)	AP	AP^{50}	AP^{75}
Large Networks								
HigherHRNet-W48 [13]	HRNet-W48	640×640	63.8	154.6G	–	65.9	86.4	70.6
DEKR [17]	HRNet-W48	640×640	65.7	141.5G	–	67.3	86.4	72.2
SWAHR [16]	HrHRNet-W48	640×640	63.8	154.6G	–	71.6	88.5	77.6
Small Networks								
HigherHRNet-W24 [13]	HRNet-W24	512×512	14.9	25.3G	–	57.4	83.2	63.2
HigherHRNet-W16 [13]	HRNet-W16	512×512	7.2	12.5G	170	50.4	78.4	54.5
EfficientHRNet [20]	EfficientHRNet-H$_{-1}$	480×480	13.0	14.2G	–	56.3	81.3	59.0
	EfficientHRNet-H$_{-3}$	416×416	5.3	4.3G	132	46.1	79.3	48.3
LitePose [24]	LitePose-S	448×448	2.7	5.0G	76	58.0	80.9	61.6
	LitePose-XS	256×256	1.7	1.2G	27	49.4	74.1	51.3
WLitePose	WLitePose-S	448×448	3.6	5.8G	81	**59.6**	82.3	**63.5**
	WLitePose-XS	256×256	2.3	1.5G	29	50.6	74.8	52.8

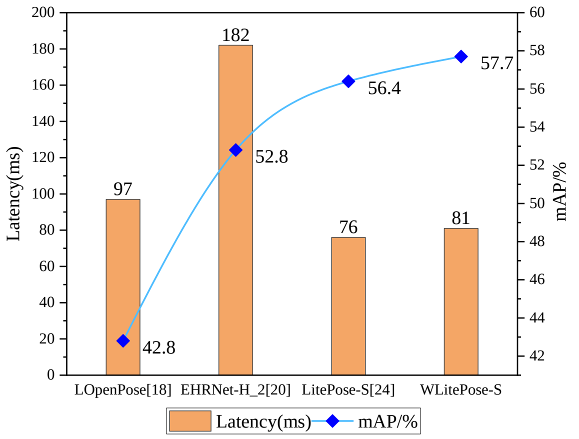

Figure 6. Latency of various lightweight networks and accuracy on COCO val2017 set

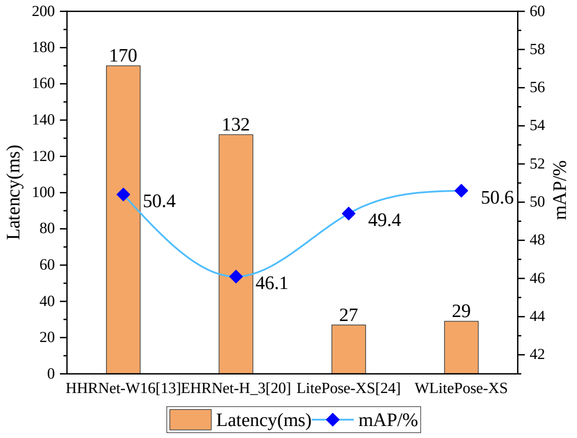

Figure 7. latency of various lightweight networks and accuracy on the CrowdPose test set

WLitePose-XS has latency nearly identical to LitePose-XS, significantly lower than that of HigherHRNet-W16 and EfficientHRNet-H$_{-3}$, while achieving the highest accuracy.

Due to the limited availability of bottom-up lightweight human pose estimation models, Table IV compares WLitePose with top-down lightweight models on the CrowdPose test set. To ensure fairness, the experimental setups for Lite-HRNet [22] and Dite-HRNet [23] are identical to those in this paper. The results show that regardless of whether the input image size is 448×448 or 256×256, WLitePose outperforms the others. This also indicates that the bottom-up architecture is more suitable for running on edge devices.

Table V presents the comparison results between WLitePose and top-down lightweight models on the COCO val2017 set. The experimental setups for Lite-HRNet and Dite-HRNet are identical to those in this paper. Compared to these models, WLitePose reduces MACs by 0.9G and achieves an AP score that is 4% higher than Lite-HRNet and 5.5% higher than Dite-HRNet.

E. *Ablation Study*

Ablation experiments were conducted on the CrowdPose test set to verify the effectiveness of the proposed improvements, such as the lightweight deconvolution module, heatmap weighted loss function, and DFC-bottleneck block.

Table VI shows the ablation experiment results of the

TABLE IV
COMPARISON RESULTS WITH TOP-DOWN LIGHTWEIGHT NETWORKS ON THE CROWDPOSE TEST SET

	Model	Backbone	Input Size	#Params/M	#MACs	AP	AP^{50}	AP^{75}
Top-down method	Lite-HRNet [22]	Lite-HRNet-30	448×448	1.8	7.2G	51.3	78.7	52.4
		Lite-HRNet-30	256×256	1.8	2.4G	41.8	70.6	41.2
	Dite-HRNet [23]	Dite-HRNet-30	448×448	1.8	7.2G	51.0	78.5	52.3
		Dite-HRNet-30	256×256	1.8	2.4G	41.6	70.6	40.5
Bottom-up method	WLitePose	WLitePose-S	448×448	3.6	5.8G	**59.6**	**82.3**	**63.5**
		WLitePose-XS	256×256	2.3	1.5G	50.6	74.8	52.8

TABLE V
COMPARISON RESULTS WITH TOP-DOWN LIGHTWEIGHT NETWORKS ON COCO VAL2017 SET

method	Model	Backbone	Input Size	#Params/M	#MACs	AP	AP^{50}	AP^{75}
Top-down	Lite-HRNet [22]	Lite-HRNet-30	256×256	1.8	2.4G	37.1	65.3	36.5
	Dite-HRNet [23]	Dite-HRNet-30	256×256	1.8	2.4G	35.6	63.2	34.3
Bottom-up	WLitePose	WLitePose-XS	256×256	2.3	1.5G	**41.1**	62.1	**44.3**

TABLE VI
RESULTS OF ABLATION EXPERIMENTS ON THE CROWDPOSE TEST SET

Model	DFC-bottleneck block	lightweight deconvolution module	heatmap weighted loss function	#Params/M	#MACs	AP
Baseline				2.7	5.0G	58.0
Exp-1	√			3.3	5.5G	58.7
Exp-2	√	√		3.6	5.8G	58.8
Exp-3		√		3.0	5.2G	58.3
Exp-4			√	2.7	5.0G	59.0
Exp-5	√	√	√	3.6	5.8G	59.6

proposed model on the CrowdPose test set. The input image size is 448×448 with LitePose as the baseline. The results of Exp-1, Exp-3, and Exp-4 demonstrate that each of the three proposed improvements can effectively enhance the model's task accuracy when used independently. In Exp-1, enhancing the model's feature extraction by replacing the basic block in the backbone network led to a 0.7 percentage point increase in AP score compared to the baseline. A comparison between Exp-2 and Exp-1 shows that the introduction of the lightweight deconvolution module provides only a minimal improvement in accuracy. This may be due to the lightweight design of this module, which somewhat restricts the enhancement of accuracy. A comparison between Exp-5 and Exp-2 shows that introducing the heatmap weighted loss function increases the AP by 0.8 percentage points. This demonstrates the effectiveness of the method, as the model pays more attention to the regions that are more valuable for keypoint localization during the training process.

To assess the impact of placing the DFC-bottleneck block at different positions within the backbone on accuracy, comparison experiments were conducted on the CrowdPose test set, as shown in Table VII. The input image size was set to 448×448, with LitePose used as the baseline. The backbone is divided into four stages based on feature size. Replacing the basic block in any stage of the backbone improves task accuracy. When the basic blocks in all four stages are replaced, the highest task accuracy is achieved while maintaining low computational complexity and parameter count. Therefore, this study replaces the basic blocks in all four stages with the proposed DFC-bottleneck block.

TABLE VII
COMPARISON EXPERIMENT OF DFC-BOTTLENECK BLOCK IN DIFFERENT POSITIONS OF BACKBONE

Stage	#Params/M	#MACs	AP
Baseline	2.7	5.0G	58.0
Stage 1	2.9	5.2G	58.2
Stage 2	3.0	5.2G	58.3
Stage 3	3.0	5.3G	58.2
Stage 4	3.1	5.3G	58.5
All	3.3	5.5G	58.7

TABLE VIII
THE IMPACT OF THE HYPERPARAMETER VALUES IN THE HEATMAP WEIGHTED LOSS FUNCTION ON TASK ACCURACY

Model	heatmap weighted loss function	hyperparameter τ	AP
Baseline			58.0
Exp-1	√	1.0	58.5
Exp-2	√	0.1	58.8
Exp-3	√	0.01	59.0
Exp-4	√	0.001	59.0

In the heatmap weighted loss function, τ is a hyperparameter that controls the position of the soft boundary. As τ decreases, the threshold θ also decreases exponentially. To evaluate the effect of different τ values on task accuracy, comparison experiments were performed on the CrowdPose test set, as shown in Table VIII. The input image size was set to 448×448, with LitePose serving as the baseline. The AP scores are identical when τ is set to 0.01 and 0.001, both achieving the highest accuracy of 59 percentage points. When τ is set to 0.01, the threshold θ is approximately 8×10^{-31}, at which point the heatmap values of many pixels in the ground truth heatmap will be greater than θ. Thus, a value of 0.01 for τ is sufficient, and there is no need to decrease it further.

The proposed DFC-bottleneck block utilizes the DFC [32] attention module. To validate the rationale for selecting the DFC attention module, the impact of employing different attention modules within the DFC-bottleneck block on task accuracy was evaluated, as shown in Table IX. Compared to the SE [33] and CBAM [34] modules, the use of the DFC attention module results in minimal differences in parameter count and computational load, but significantly improves accuracy. Compared to the CA [35] module, although both achieve the same AP score, the DFC attention module has a lower computational load, making it more advantageous. The results fully validate the rationale and superiority of using the DFC attention module within the DFC-bottleneck block.

To assess the impact of each enhancement in the DFC

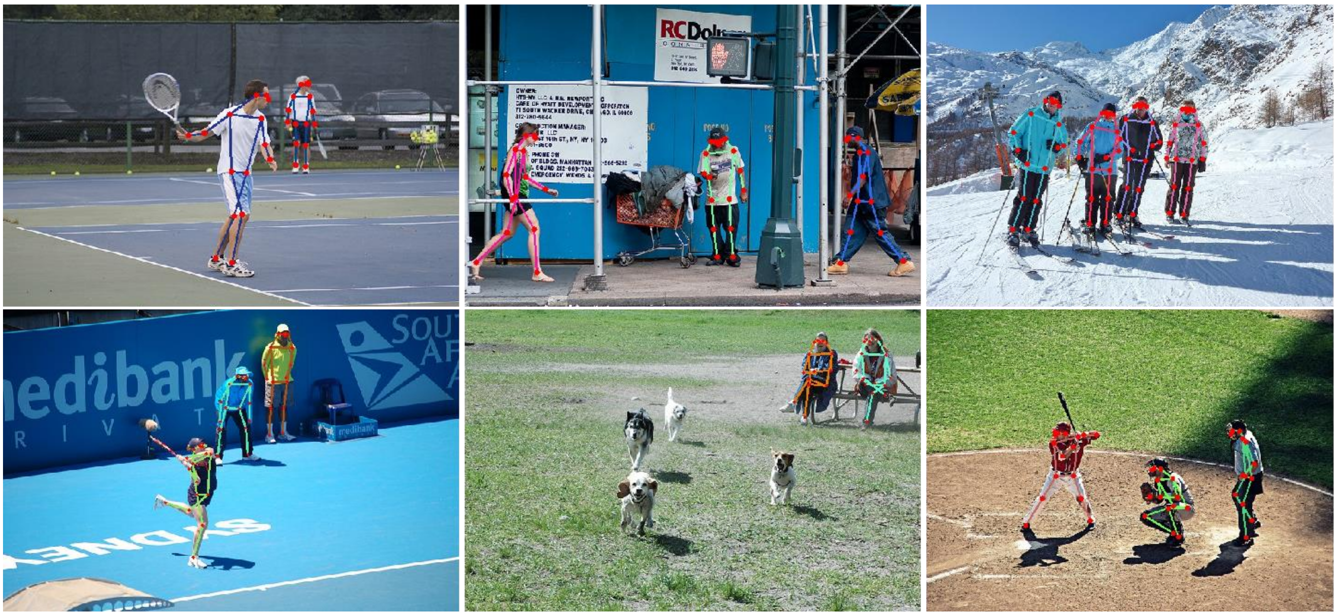

Figure 8. Visual results of pose estimation on the COCO val2017 set

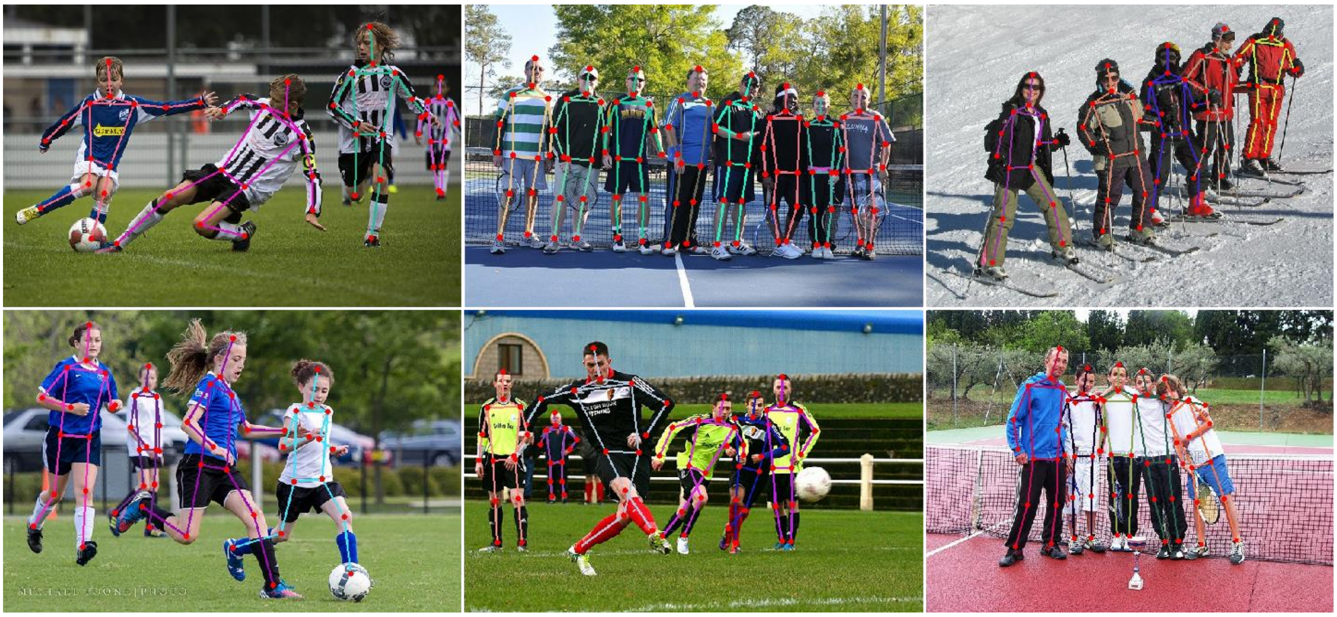

Figure 9. Visual results of pose estimation on the CrowdPose test set

TABLE IX
THE IMPACT OF USING DIFFERENT ATTENTION MODULES WITHIN THE DFC-BOTTLENECK BLOCK ON TASK ACCURACY

Model	attention module	#Params/M	#MACs	AP
Baseline		2.7	5.0G	58.0
Exp-1	SE [33]	2.9	5.3G	58.2
Exp-2	CBAM [34]	2.9	5.3G	58.3
Exp-3	CA [35]	3.2	5.7G	58.7
Exp-4	DFC [32]	3.3	5.5G	58.7

TABLE X
EXPERIMENTAL DETAILS OF THE DFC-BOTTLENECK BLOCK DESIGN

Model	#MACs	AP
Baseline	5.0G	58.0
move up 7×7 depthwise conv	4.6G	57.1
+ DFC attention	5.5G	58.4
ReLU → GELU	5.5G	58.4
fewer activations	5.5G	58.6
fewer batch norms	5.5G	58.7

bottleneck block on the model's performance, relevant experiments were carried out. Table X presents the experimental details of the block's design. Initially, moving the 7×7 depthwise convolutional layer forward resulted in a 0.9% drop in AP score compared to the baseline but reduced the computational load. Then, adding the DFC attention mechanism in parallel with the 7×7 depthwise convolution led to a 0.4% AP score improvement over the baseline, with minimal increase in computational load. Next, the activation function was changed from ReLU to GELU, and the number of GELU and BN layers was reduced. These adjustments formed the DFC-bottleneck block, which, compared to the baseline, only increased MACs by 0.5G while improving the AP score by 0.7%. These results validate the effectiveness and advantages of the DFC-bottleneck block design.

F. *Visualization Results*

Figure 8 illustrates the pose estimation visualization on the COCO val2017 dataset. WLitePose exhibits accurate detection of keypoints for smaller individuals or those in the background, even in cases of body part overlap or partial occlusion. Figure 9 showcases the pose estimation visualization on the CrowdPose test set. In more complex scenarios, multi-person pose estimation encounters difficulties like occlusion, scale changes, blur, and background distractions. Despite these challenges, the method retains strong detection performance, highlighting its robustness.

V. Conclusion

This paper introduces WLitePose, a lightweight bottom-up human pose estimation model based on the LitePose architecture. The study revisits the shortcomings of the MSE loss function and presents a novel heatmap weighted loss function. This approach allows the model to pay more attention to the regions that are more valuable for keypoint localization during the training process. Additionally, a lightweight deconvolution module is incorporated into the main architecture to produce higher-resolution heatmaps. During inference, these heatmaps are aggregated to improve keypoint prediction for smaller individuals, thereby strengthening the model's capacity to manage variations in human scale. To boost feature extraction in the backbone, the basic block is substituted with the newly proposed DFC-bottleneck block, which increases overall model accuracy. WLitePose achieves strong performance on public datasets. This model strikes an effective balance between accuracy and efficiency, broadening its potential for practical applications.

References

[1] Andi W. R. Emanuel, Paulus Mudjihartono, and Joanna A. M. Nugraha, “Snapshot-Based human action recognition using OpenPose and deep learning,” IAENG International Journal of Computer Science, vol. 48, no. 4, pp. 862-867, 2021.

[2] Nha Tran, Hung Nguyen, Hien Luong, Minh Nguyen, Khiet Luong, and Huy Tran, “Recognition of student behavior through actions in the classroom,” IAENG International Journal of Computer Science, vol. 50, no. 3, pp. 1031-1041, 2023.

[3] N. Yang, J. Zhao, “Dangerous driving behavior recognition based on improved YoloV5 and Openpose,” IAENG International Journal of Computer Science, vol. 49, no. 4, pp. 1112-1122, 2022.

[4] Y. Guan, and W. Mao, “Pedestrian virtual space based abnormal behavior detection,” IAENG International Journal of Computer Science, vol. 46, no. 2, pp. 311-320, 2019.

[5] G. Sateesh Babu, B.S. Mahanand, “Computer aided detection of imaging biomarkers for Alzheimer's disease,” International Journal of Signal and Imaging Systems Engineering, vol. 12, no. 3, pp. 108-118, 2021.

[6] Shilpa Rani, Kamlesh Lakhwani, Sandeep Kumar, “Syntactic approach to reconstruct simple and complex medical images,” International Journal of Signal and Imaging Systems Engineering, vol. 12, no. 4, pp. 127-136, 2023.

[7] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” Proceedings of the European Conference on Computer Vision, pp. 472-487, 2018.

[8] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, “CrowdPose: Efficient crowded scenes pose estimation and a new benchmark,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10855-10864, 2019.

[9] K. Sun, B. Xiao, and D. Liu, “Deep high-resolution representation learning for human pose estimation,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5686-5696, 2019.

[10] H. -S. Fang et al., “AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 6, pp. 7157-7173, 2023.

[11] A. Newell, Z. Huang, and J. Deng, “Associative embedding: End-to-end learning for joint detection and grouping,” Proceedings of the Conference on Neural Information Processing Systems, pp. 2278-2288, 2017.

[12] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, “PersonLab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model,” Proceedings of the European Conference on Computer Vision, pp. 282-299, 2018.

[13] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, “HigherHRNet: Scale-aware representation learning for bottom-up human pose estimation,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5385-5394, 2020.

[14] M. Y. Li, and J. Zhao, “CE-HigherHRNet: Enhancing channel information for small persons bottom-up human pose estimation,” IAENG International Journal of Computer Science, vol. 49, no. 1, pp. 260-269, 2022.

[15] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” Proceedings of the European Conference on Computer Vision, pp. 483-499, 2016.

[16] Z. Luo, Z. Wang, Y. Huang, L. Wang, T. Tan, and E. Zhou, “Rethinking the heatmap regression for bottom-up human pose estimation,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13259-13268, 2021.

[17] Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang, “Bottom-up human pose estimation via disentangled keypoint regression,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14671-14681, 2021.

[18] Daniil Osokin, “Real-time 2d multi-person pose estimation on cpu: Lightweight openpose,” arXiv preprint arXiv:1811.12004, 2018.

[19] L. Xu, Y. Guan, S. Jin, W. Liu, C. Qian, P. Luo, W. Ouyang, and X. Wang, “Vipnas: Efficient video pose estimation via neural architecture search,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16067-16076, 2021.

[20] Christopher Neff, Aneri Sheth, Steven Furgurson, and Hamed Tabkhi, “Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation,” arXiv preprint arXiv:2007.08090, 2020.

[21] M. Tan, Quoc V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” arXiv preprint arXiv:1905.11946, 2019.

[22] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang, “Lite-hrnet: A lightweight high-resolution network,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10435-10445, 2021.

[23] Q. Li, Z. Zhang, F. Xiao, F. Zhang and Bir Bhanu, “Dite-HRNet: Dynamic lightweight high-resolution network for human pose estimation,” arXiv preprint arXiv: 2204.10762, 2022.

[24] Y. Wang, M. Li, H. Cai, W. Chen and S. Han, “Lite Pose: Efficient architecture design for 2D human pose estimation,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13116-13126, 2022.

[25] C. Du, Z. Yan, H. Yu, L. Yu and Z. Xiong, “Hierarchical associative encoding and decoding for bottom-up human pose estimation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 4, pp. 1762-1775, 2023.

[26] L. Jin, X. Wang, X. Nie, L. Liu, Y. Guo and J. Zhao, “Grouping by center: Predicting centripetal offsets for the bottom-up human pose estimation,” IEEE Transactions on Multimedia, vol. 25, pp. 3364-3374, 2023.

[27] H. Yu, C. Du, and L. Yu, “Scale-aware heatmap representation for human pose estimation,” Pattern Recognition Letters, vol. 154, pp. 1-6, 2022.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018.

[30] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan et al., “Searching for MobileNetV3,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314-1324, 2019.

[31] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6848-6856, 2018.

[32] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, Y. Wang, "GhostNetV2: Enhance cheap operation with long-range attention," arXiv preprint arXiv:2211.12905, 2022.

[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," Proceedings of the European Conference on Computer Vision, pp. 3-19, 2018.

[35] Q. Hou, D. Zhou and J. Feng, "Coordinate attention for efficient mobile network design," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13708-13717, 2021.

[36] Y. Xu, J. Zhang, Q. Zhang and D. Tao, "ViTPose++: Vision transformer for generic body pose estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 2, pp. 1212-1230, 2024.

[37] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal loss for dense object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 2020.

[38] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," Proceedings of the European Conference on Computer Vision, pp. 740-755, 2014.

[39] Z. Liu, H. Mao, C. -Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, "A ConvNet for the 2020s," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11966-11976, 2022.