2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

MPPose: An Efficient Multi-Path Network for 2D Human Pose Estimation

Qing Peng[1][0009-0009-9767-1671], Zhongteng Zhang[1][0009-0003-9427-1126], Zihao Zhang[4], Liu Zhang[1][0009-0005-2471-6147], Jing Chong[5] and Weihong Huang[2,3,4]✉[0000-0002-7168-4943]

[1] Central South University, School of Computer Science and Engineering, Changsha, China
[2] Xiangya Hospital of Cnetral South University (National Clinical Research Center for Geriatric Disorders (Xiangya Hospital)), Changsha, China
[3] Mobile Health Ministry of Education - China Mobile Joint Laboratory, Changsha, China
[4] Central South University, Big Data Insititute, Changsha, China
[5] China Mobile (Chengdu) Industrial Research Institute, Chengdu, China

Abstract. Human pose estimation models are increasingly deployed on low-computation devices, with extensive applications in motion capture and sports rehabilitation. The multi-scale feature extraction capability of high-resolution networks (HRNet) effectively addresses the issue of varying human body scales, enhancing the accuracy of lightweight models based on HRNet. However, the high-resolution architecture results in a more complex network structure and increased computational overhead. This paper introduces MPPose, a top-down human pose estimation framework that integrates coordinate classification based on keypoint heatmap representation. We design a single-branch network based on a high-resolution architecture, which implicitly retains and fuses multi-scale features. The multi-path network maintains both the simplicity of single-branch network and the effectiveness of high-resolution network, resulting in a simpler and more efficient architecture.

Based on the high-resolution architectures, we retain only the blocks in the lowest-resolution branch and employ both cross-resolution and same-resolution feature fusion. We redesign an efficient block inspired by the shuffle block, which we called the Channel Expansion Attention Module (CEAM). CEAM compensates for the reduction in channel information caused by channel splitting by introducing a channel scaling module and a channel attention module. We evaluate our model against state-of-the-art top-down methods on the COCO and MPII datasets. Results show that it reduces computational overhead by 20% and improves inference speed by 37%, while achieving accuracy on par with Lite-HRNet.

Keywords: 2D Human Pose Estimation, Lightweight Network, Efficient Block.

1 Introduction

Human pose estimation aims to predict keypoints of interest from an image and assemble them into a representation of the human pose. It has broad applications in fields such as sports monitoring and medical rehabilitation training [1-3]. Typical human pose estimation can be categorized into two paradigms: top-down [4-9] and bottom-up [6,10-12]. The top-down paradigm first detects people in an image via a person detector, and then performs single-person pose estimation for each detected person. These methods benefit from straightforward operations and the excellent performance of human detectors, enabling higher accuracy. In contrast, the bottom-up paradigm directly predicts all human keypoints and then groups these keypoints to obtain each person's keypoints. Although current research tends to focus on end-to-end approaches, the top-down method is simpler and can achieve higher accuracy, making it more suitable for low-density crowd scenes and the design of lightweight human pose estimation models [13-15].

Human pose estimation study based on high-resolution networks typically requires high-resolution representations to achieve high performance [7,10-18], resulting in high computational complexity and low inference speed. Existing studies mainly desgin lightweight networks from two perspectives. One is to optimize the efficient blocks [18,19], such as MobileNet [20-22] and ShuffleNet [23,24]. These efficient blocks typically adopt unique approaches to reduce convolution operations with minimal performance sacrifice. The other is to use attention mechanisms to focus on important features, which can significantly enhance performance. Lite-HRNet [18] uses the proposed conditional channel weighting block to replace the residual block in HRNet [7]. They employed cross-resolution weight and spatial weight in the Shuffle block to replace 1×1 convolution, aiming to reduce the computational complexity of the block. However, this complex weight calculation is detrimental to inference speed, and no modifications are made to the high-resolution network structure.

In fact, computation in high-resolution network is primarily concentrated on the high-resolution branches, but some studies [19,25] have shown that high resolution architecture is not necessary for lightweight network. To focus computational resources on high-level semantics, we first remove most of the blocks from Lite-HRNet. Specifically, we remove all blocks except for the blocks in the lowest resolution branch in each stage, resulting in a single-branch structure. We still retain feature fusion operations between different branches within each stage to benefit from high-resolution representations. In addition to using cross-resolution feature fusion, we also fuse the features within the same-resolution branch. In high-resolution networks, as the resolution decreases, the number of channels increases, but the model cannot always extract more information from the additional channels, making an excessive number of channels unnecessary. Therefore, following the principle proposed by ShuffleNet, we perform a channel split on each branch to reduce the number of channels. We split each branch into two parts based on the number of channels. One part is used to fuse cross-resolution features, while the other part is used to fuse same-resolution features in the subsequent stage.

Attention mechanisms [26-28] in high-resolution networks are primarily employed within the same-resolution branch or the same stage. In Lite-HRNet, both spatial weight and cross-resolution weight are used to replace 1×1 convolution in the shuffle block. Although this method significantly improves the performance, complex weight computation impacts the model's inference speed. In our work, we add a basic channel attention block [26] to the original shuffle block to focus on important features. We believe that the channel split, which halves the number of channels, becomes a bottleneck for the performance improvement brought by basic channel attention. To further enhance the efficiency of the proposed basic block, two convolution layers in the shuffle block are used to expand and reduce the number of channels. The 3×3 convolution operates on the expanded feature map, and the channel attention block is then used to compute weights between channels for channel scaling. The expansion and shrinking operations significantly enhance the performance of the channel attention block in the low-channel shuffle block structure, and we call the proposed block as CEAM. By replacing the basic blocks in the proposed network structure with CEAM, we obtain a new simple and efficient network, called MPPose.

Results show that the proposed MPPose outperforms ShuffleNet, MobileNet and Lite-HRNet. We believe that the superiority of our model is attributed to the proposed network structure, which is more aligned with the demands of lightweight network design. The proposed CEAM significantly improves efficiency and speed compared to mainstream efficient blocks.

Our main contributions include:

- We redesign the high-resolution network structure by removing the blocks in high-resolution branches, resulting in a new multi-path network. This structure retains the advantages of a single-branch network while benefiting from both cross-resolution and same-resolution feature fusion.
- We propose a simple and efficient CEAM block, which is more suitable for lightweight networks. CEAM compensates for the performance bottleneck caused by the channel splitting in the shuffle block by introducing channel scaling and channel attention, further improving the performance of the efficient block.
- Experiments on the COCO and MPII datasets demonstrate the effectiveness of our method. Compared to the advanced Lite-HRNet model, our model achieves approximately a 20% reduction in computational complexity and a 37% improvement in inference speed.

2 Related Work

2.1 Top-down Human Pose Estimation

human pose estimation aims at identifying the positions of keypoints on the human body to determine the action and pose. Typical top-down approaches decompose the task into two processes: first, a person detector is used to identify each person in the image, and then a single-person pose estimation is performed for each individual. Among these methods, HRNet [7] achieve better performance than single-branch architectures. HRNet designed a multi-branch architecture to allow multi-resolution fusion, which has been proven effective in solving scale variation problems. However, it significantly increases the computational demands, making it less suitable for deployment on mobile devices and challenging to support real-time applications. Some lightweight HRNet-related studies [10,18,29] such as Lite-HRNet [18] proposed a more efficient and simpler module combined with attention mechanism (incorporating spatial and Cross-resolution weighting information). However, these complex attention mechanisms are not computationally friendly. In this work, we redesign a simpler network and module based on Lite-HRNet and replace heatmap-based regression method with computationally-friendly coordinate regression method.

2.2 Efficient CNN Blocks

Previous studies have introduced group convolutions and depthwise separable convolutions, which have gained increasing attention in the design of lightweight models, such as Xception [30], MobileNet, and ShuffleNet. Many studies have demonstrated that these lightweight convolution operations are effective, significantly reducing computation with little performance decrease, and now widely used in the design of lightweight networks [31]. ShuffleNetV2 proposed an efficient shuffle block and explored four principles for designing lightweight modules. Lite-HRNet proposed a new conditional channel weighting block based on shuffle block, which replaced the costly 1×1 convolution with spatial weight and cross-resolution weight, similar to attention mechanisms. However, the conditional channel weighting block is complex, and the channel split significantly reduces the number of channels, making information exchange across channels challenging. In our work, we use 1×1 convolution to adjust the number of channels. The primary feature extraction operations are performed after channel expansion, and we employ a channel attention module to capture important channel features, making it more efficient in leveraging the benefits of channel expansion.

2.3 Single Branch and High-Resolution

Single-branch network structures [32-34] have long been the mainstream structure for lightweight network design. These networks have a simple structure, consisting of a main upsampling and downsampling path, forming an encoder-decoder structure. With the introduction of high-resolution networks, the performance of human pose estimation models has significantly improved. This structure maintains multiple high-resolution paths, effectively addressing the issue of scale variation, and significantly improving performance compared to single-branch networks. Recent studies have started to focus on designing lightweight networks based on high-resolution architecture. However, lightweight studies focusing on high-resolution architectures are still limited to multi-branch structures. Some previous studies have indicated that high-resolution branches in high-resolution architectures are redundant for lightweight models, and reallocating computational resources to low-resolution branches can yield greater performance improvements. In our work, we revisit high-resolution networks and design a single-branch network that integrates both cross-resolution and same-resolution feature fusion, achieving better performance and efficiency.

3 Method

3.1 Structure

The basic architecture of our proposed MPPose is presented in Fig. 1. We will introduce an overview of our model and our design philosophy. We first take Lite-HRNet as our basic backbone network, which redesigned from small HRNet. The original high-resolution network comprises four parallel branches, each performing substantial computational operations, resulting in significant computational overhead. To address this issue, we remove the computational operations from the high-resolution branches at each stage, retaining only the lowest resolution branch's operations. However, to maintain the ability to handle scale variation, we preserve the feature fusion operations across different resolution branches. This result in our model's main structure: a single-branch structure that integrates multi-resolution features.

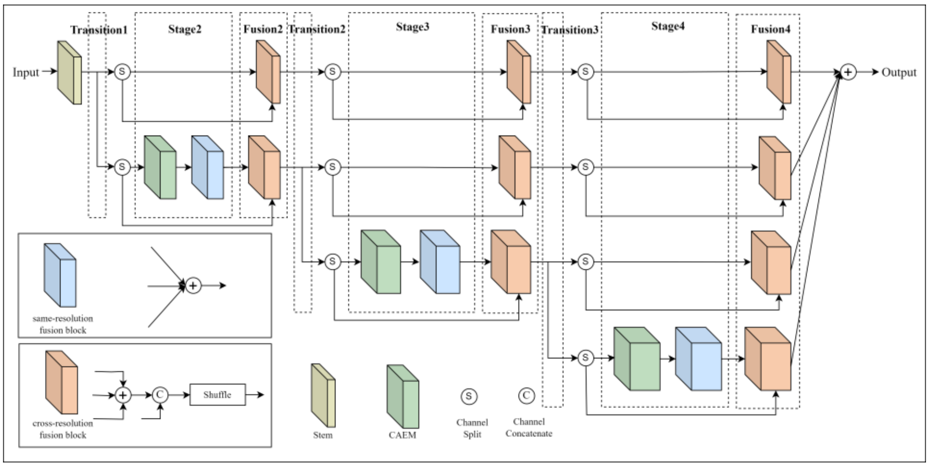

Fig. 1. The architecture of MPPose. MPPose consists of four parallel branches, including stem layer, transition layer, stage layer, and fusion layer. The transition layer is responsible for generating new low-resolution branches. The stage layer comprises the core operations of the network, including CEAM block and cross-resolution feature fusion. The fusion layer is designed for same-resolution feature fusion.

To reduce the model's parameters and computational cost, we employ the CEAM as the main feature extraction block, which is redesigned from the Shuffle block. The whole network can be divided into three parts: stem, backbone, and head. The stem consists of a 3×3 convolution followed by a basic shuffle block, both downsampling with a stride of 2, reducing the feature map resolution by a factor of four, expanding the number of channels from 3 to 32 and 128, respectively.

The backbone can be divided into three stages, each with 2, 3, and 4 branches, respectively. The structure of the branches remains consistent with the HRNet design, where the lower-resolution branches are downsampled from higher-resolution branches, with the channel doubling at each stage. However, in our model's backbone, each stage retains blocks only on the lowest resolution branch for feature extraction, while the other branches are not processed and are used only for feature interaction. In addition, the high-resolution network includes cross-resolution feature fusion, to which we add same-resolution feature fusion. Specifically, in each stage, we initially split the branches on channels and then concatenate and shuffle them at the end of the branch.

Ultimately, each branch retains original features and fusion with other branches, effectively adding extra residual connections throughout the structure. These residual connections preserve original features while reducing model complexity, which is a common practice in lightweight network design. The cross-resolution fusion block represents the feature fusion between the current branch and other branches, similar to the operation in the high-resolution network. The same-resolution feature fusion block first fuses the features split in the previous stages, then performs channel concatenation and shuffling operations with the features split in the current stage to fully fuse the same-resolution features. Table 1 shows the detailed structure of our proposed MPPose.

Table 1. Structure of MPPose. The resolution branch indicates the number of branches within the current module and their respective downsampling ratios relative to the original input resolution. The repeat denotes the number of times each operation is repeated, while the modules specifies the number of times each module is repeated.

Layers	operator	Resolution branch	repeat	modules
image		1×		
stem	Conv2d	2×	1	1
	Shuffle Block	4×	1	
stage2	CEAM Block	4× 8×	3	2
	Cross-resolution Fusion Block	4× 8×	1	
fusion2	Same-resolution Fusion Block	4× 8×	1	1
stage3	CEAM Block	4× 8× 16×	3	4
	Cross-resolution Fusion Block	4× 8× 16×	1	
fusion3	Same-resolution Fusion Block	4× 8× 16×	1	1
stage4	CEAM Block	4× 8× 16× 32×	3	2
	Cross-resolution Fusion Block	4× 8× 16× 32×	1	
fusion4	Same-resolution Fusion Block	4× 8× 16× 32×	1	1
FLOPs				0.16G
Params				4.2M

We finally use proposed regression-based method [35] as the head for our model. The output of the backbone is processed through two linear layers to predict the horizontal coordinate and vertical coordinate, respectively. First, the feature map from each channel is flattened into a one-dimensional vector. These vectors are then passed through two linear layers for prediction, resulting in horizontal coordinate and vertical coordinate. The horizontal coordinate and vertical coordinate for the same keypoint are combined to obtain the final coordinate. Compared to directly outputting a heatmap, this method does not require post-processing, allowing for direct keypoint coordinate extraction, which provides a speed advantage.

3.2 CEAM Module

Many lightweight blocks have been proposed, benefiting from their efficient lightweight design concepts and excellent performance [36,37], increasing the variety of lightweight modules available. ShuffleNetV2 is a classic efficient block that proposed four principles for designing efficient CNN architecture. It introduces the ShuffleNetV2 Block, as shown in Fig. 2(a), based on these principles. Its basic unit consists of two paths: first, the feature map split into two branches via channel split, one branch remains unprocessed, while the other passes through a 1× 1 convolution, 3×3 depthwise convolution, and another 1×1 convolution sequentially. Finally, the two parts undergo channel concatenation and shuffle. In this process, the unprocessed branch reduces the overall computation and parameters of the module while retaining some original feature information through residual connections. The success of the Shuffle Block is due to channel split, but this also leads to a reduction in the number of channels, causing a performance bottleneck.

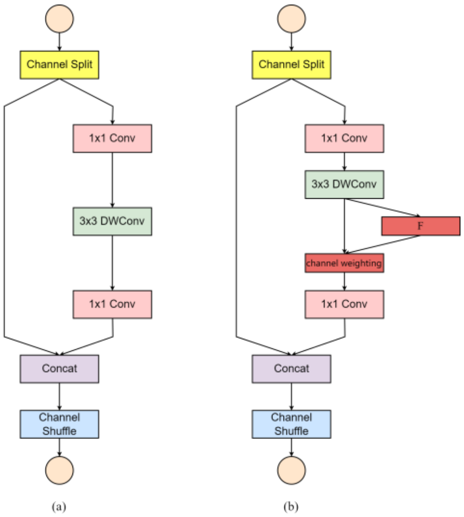

Fig. 2. Building Blocks. (a) The shuffle block. (b) Our proposed CEAM. Two 1×1 convolutions are used to expand and reduce the number of channels. F = channel weighting function.

Our module introduces channel expansion on this basic shuffle block, using the original 1×1 convolution to expand and shrink the number of channels. The 3×3 convolution is performed on the expanded part, and channel attention is introduced to more efficiently utilize the benefits brought by channel expansion. The proposed block, called the Channel Expansion Attention Module (CEAM), is illustrated in **Fig. 2**(b). The first 1×1 convolution expands the number of channels to five times the original, and then the 3×3 depthwise convolution extracts feature on the expanded feature map. Finally, the second 1×1 convolution reduces the channel back to the original size. This process forms an inverted bottleneck structure similar to that proposed in MobileNetV2, which has been shown to be effective in their results. Additionally, we consider the limitations of simple channel expansion operations and use a channel attention to focus on important channel features, fully utilizing the benefits brought by channel expansion. Attention modules focus on important parts or features of the input data to enhance model performance and effectiveness. Considering the complexity of attention mechanisms, we use basic channel attention module that only includes global pooling and two linear layers. After max pooling to obtain the average along the channel dimension, two linear layers compress and expand dimensions, and an activation function generates weights to weight the original feature map, producing the final feature map. Formula (1) shows the calculation formula for basic attention function.

$$
\begin{gathered}
z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{c,i,j} \\
F(X) = \sigma(W_2 \delta(W_1 z)) = \sigma(W_2 \delta(W_1 z_c))
\end{gathered}
\tag{1}
$$

4 Experiments

4.1 Dataset & Metrics

COCO. COCO [39] contains over 200K images and 250K person instances with 17 keypoints, and it is divided into train, val, test-dev sets. We trained our model on train2017 (includes 57K images and 150K person instances) and validated it on val2017 (includes 5K images) and test-dev2017 (includes 20K images). The test-dev set is a subset of the test set. In the COCO dataset, neither the test set nor the test-dev set is publicly labeled. They are mainly used to verify the generalization of the model on unknown datasets. The test set is mainly used for competitions, while the test-dev set can be used for development. All our experiments are trained exclusively on the COCO train set. And we report the results on COCO val set and COCO test-dev set.

MPII. MPII [40] includes full-body pose annotations taken from real-world human activities. It consists of 25K images and 40K human instances annotated with 16 keypoints, and it is divided into train/test sets with 28K person instances and 12K person instances, respectively. Unlike the COCO dataset, the MPII dataset has 16 keypoints, including the top of the head, neck, left and right shoulders, left and right elbows, left and right wrists, left and right hips, left and right knees, left and right ankles, and left and right toes. Since the MPII dataset contains the annotations of the top of the head, neck, and toes, it is more suitable for human motion analysis and can also test the model's prediction performance for asymmetric keypoints.

Evaluation Metrics. Object Keypoint Similarity (OKS) is used in the field of human pose estimation to predict the similarity between keypoints and ground-truth keypoints. It is a popular evaluation metric for current human keypoint detection algorithms and we report standard average precision and recall scores: mAP (the mean of AP scores at 10 positions, OKS = 0.50, 0.55, ..., 0.90, 0.95) on COCO. A higher OKS indicates a closer match between predicted and ground-truth keypoints. Formula (2) shows the calculation formula for OKS.

$$\mathrm{OKS}=\frac{\sum_i \exp(-\frac{d_i^2}{2s^2k_i^2})\delta(v_i>0)}{\sum_i \delta(v_i>0)} \tag{2}$$

PCKh quantifies the proportion of normalized distances between predicted keypoints and their corresponding ground-truth keypoints that fall below a specified threshold. For MPII, we use the standard metric PCKh@0.5 (head-normalized probability of correct keypoint) to evaluate the performance. Formula (3) shows the calculation formula for PCKh.

$$\mathrm{PCKh}=\frac{1}{N}\sum_{i=1}^{N}\delta(\frac{\left\|\hat{p}_i-p_i\right\|}{\mathrm{head\_sieze}}\leq\alpha) \tag{3}$$

4.2 Experiment Setting

The network is trained on a single GeForce RTX 4090D GPU with min-batch size of 128, and the speed experiments are tested in RTX 3090. All experiments adopt Adam optimizer with an initial learning rate of 1e-3, with epochs set to 210. The learning rate was reduced to 1e-4 and 1e-5 at the 170th and 200th epochs, respectively. The human detection boxes were expanded to a fixed 4:3 ratio before cropping from the image. The image sizes for the COCO dataset were adjusted to 256×192 or 384×288, and 256×256 for the MPII dataset. Consistent with HRNet and Lite-HRNet, data augmentation includes random rotation ([-30,30]), random scale [0.75,1.25], random translation ([-40,40]), and random flip.

As a two-stage top-down approach [41], we first used a human detector to identify human instances, followed by keypoint prediction. The human detector used on the COCO dataset was consistent with that of HRNet and Lite-HRNet. For the MPII dataset, the provided person boxes were used, following standard testing strategies to comparison with other methods. Specifically, we used simple coordinate regression to predict keypoints, predicting the x and y coordinates to get the keypoint positions in the image.

4.3 Experiments Results

COCO val. The results of our method compared to other state-of-the-art methods are reported in Table 2.

Table 2. Comparisons of various networks on the COCO val set. Params and GFLOPs are calculated for the pose estimation network, GFLOPs is for convolution and linear layers only.

Model	Input Size	Params	GFLOPs	mAP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Hourglass	256×192	25.1M	14.3	66.9	-	-	-	-	-
CPN	256×192	27.0M	6.20	68.6	-	-	-	-	-
SimpleBaseline	256×192	34.0M	8.90	70.4	88.6	78.3	67.1	77.2	76.3
HRNet	256×192	28.5M	7.10	73.4	89.5	80.7	70.2	80.1	78.9
DARK [38]	128×96	63.6M	3.60	71.9	89.1	79.6	69.2	78.0	77.9
MobileNetV2	256×192	9.6M	1.48	64.6	87.4	72.3	61.6	71.2	70.7
MobileNetV2	384×288	9.6M	3.33	67.3	87.9	74.3	62.8	74.7	72.9
ShuffleNetV2	256×192	7.6M	1.28	59.9	85.4	66.3	56.6	66.2	66.4
ShuffleNetV2	384×288	7.6M	2.87	63.6	86.5	70.5	59.5	70.7	69.7
Lite-HRNet	256×192	1.1M	0.20	64.8	86.7	73.0	62.1	70.5	71.2
Lite-HRNet	384×288	1.1M	0.45	67.6	87.8	75.0	64.5	73.7	73.7
MPPose(ours)	256×192	4.2M	**0.16**	**64.8**	86.5	72.6	61.8	70.8	70.8
MPPose(ours)	384×288	10.7M	**0.37**	**67.7**	87.9	75.1	64.7	73.8	74.3

Our MPPose, trained with an input size of 256×192, achieved an AP score of 64.8, the same as Lite-HRNet-18 and outperforms other models with fewer GFLOPs. MPPose achieves the same performance with only 80% of the GFLOPs of Lite-HRNet-18. Compared to MobileNetV2 and ShuffleNetV2, MPPose achieves accuracy improvements of 0.2 and 4.9 points, respectively, with only 11% and 13% of their computational complexity. Compared to some large network, such as Hourglass, CPN, SimpleBaseline, HRNet and DARK, MPPose achieves comparable AP score with far low complexity. When trained with an input size of 384×288, MPPose achieved an AP of 67.7, surpassing Lite-HRNet and other light-weight methods.

To compare the inference speed between MPPose and Lite-HENet, we measured the total time taken to predict keypoints on the COCO validation set. Fig. 3 compares the Lite-HRNet with our proposed MPPose, demonstrating that MPPose achieves higher speed while maintaining the same performance. MPPose achieves comparable performance to Lite-HRNet while utilizing only 80% of the GFLOPs and improves inference speed by 37%. This indicates that MPPose's multi-path network structure achieves a better balance between lightweight design and performance. Compared to Lite-HRNet, MPPose reduces computational operations in the high-resolution branches, focusing the limited computational resources on extracting high-level semantic information. Furthermore, multi-scale feature fusion allows the model to consider both global information and local details, resulting in more accurate keypoint representations with core predictive features. Additionally, the simple coordinate classification method replaces the heatmap-based post-processing, improving the speed of keypoint localization and significantly enhancing the overall prediction speed of the model.

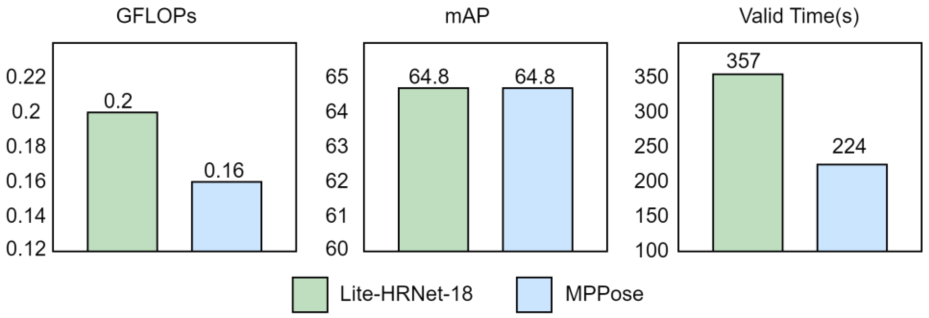

Fig. 3. Compared with the Lite-HRNet-18 on GFLOPs, mAP and valid time. The valid time is tested on COCO val use RTX 3090. The valid time is compared based on the total time taken from inputting the model to obtaining the keypoint coordinate JSON file for all images in the COCO val set using the detection box, with a total of 104,125 samples.

To analyze the actual prediction performance of the MPPose model, the paper extracts some prediction results on the COCO validation set and visualizes the predicted keypoint coordinates on the images. From the prediction visualization in Fig. 4, it can be seen that the MPPose predictions closely match the actual keypoint locations, especially when the human body occupies a significant portion of the image, where the results are even more accurate. This reflects the typical scenario in real-world applications, indicating the feasibility and effectiveness of MPPose in practical use.

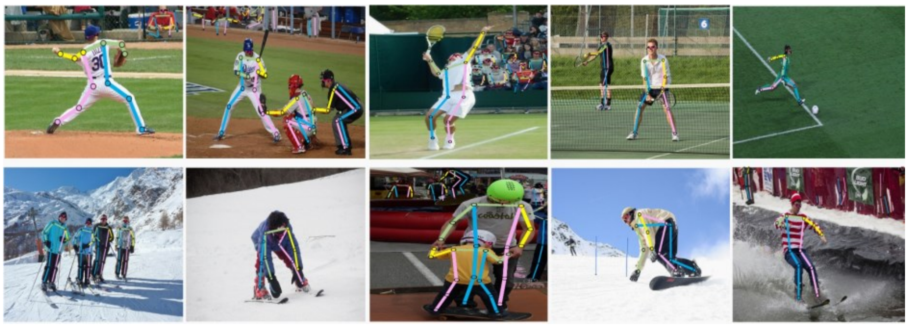

Fig. 4. The model visualizes results on the COCO val set. The visualizations demonstrate that our model has an advantage in addressing scale variations of human body.

COCO test-dev. The results of our method compared to other state-of-the-art methods in COCO test-dev are reported in Table 3. With an image input size of 384×288, our MPPose achieves an AP score of 66.9 with only 0.37 GFLOPs, the same as Lite-HRNet-18 and outperforms other models. We also test the image input size of 256×192, our MPPose achieves 63.8 AP with 0.16GFLOPs, even outperforming ShuffleNetV2 and Small HRNet with input size of 384×288.

Table 3. Comparisons of various networks on the COCO test-dev set. Params and FLOPs are calculated for the pose estimation network, GFLOPs is for convolution and linear layers only.

Model	Input Size	Params	GFLOPs	mAP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Mask-RCNN	-	-	-	63.1	87.3	68.7	57.8	71.4	-
CPN	384×288	-	-	72.1	91.4	80.0	68.7	77.2	78.5
SimpleBaseline	384×288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNet	384×288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
DARK	384×288	63.6M	32.9	76.2	92.5	83.6	72.5	82.4	81.1
MobileNetV2	384×288	9.8M	3.33	66.8	90.0	74.0	62.6	73.3	72.3
ShuffleNetV2	384×288	7.6M	2.87	62.9	88.5	69.5	58.9	69.3	68.9
Lite-HRNet	384×288	1.1M	0.45	66.9	89.4	74.4	64.0	72.2	72.6
MPPose(ours)	256×192	4.2M	0.16	64.3	88.8	71.6	61.5	69.5	70.2
MPPose(ours)	384×288	10.7M	**0.37**	**66.9**	89.7	74.5	63.7	72.6	73.3

MPII val. We also report the results on MPII dataset in Table 4. The performance of our method is similar to Lite-HRNet and other prior light-weight methods with fewer GFLOPs. our MPPose achieve 83.3 PCKh@0.5 with only 0.22GFLOPs. MPPose achieves performance close to other lightweight human pose estimation models on the MPII dataset, with a PCKh of 83.3%. However, there is still a noticeable gap compared to current state-of-the-art models. This can be attributed to two main factors: first, the relatively small size of the MPII dataset limits the model's ability to learn image features effectively during cross-resolution and same-resolution feature fusion. Second, the MPII dataset contains color-related features due to human clothing, which interact with similar environments and objects. These irrelevant features may be learned by the model, introducing erroneous information that results in a performance decline during the final prediction.

Table 4. Comparisons of various networks on the MPII val set. The image input size for all networks is 256×256, GFLOPs is for convolution and linear layers only.

Model	Params	GFLOPs	PCKh
MobileNetV2	9.6M	1.97	85.4
MobileNetV3	8.7M	1.82	84.3
ShuffleNetV2	7.6M	1.70	82.8
Small HRNet	1.3M	0.72	80.2
Lite-HRNet	1.1M	0.27	86.1
MPPose(ours)	5.6M	0.22	83.3

4.4 Ablation Study

CEAM Block. As shown in Table 5, without using channel expansion and channel attention, the module is the original Shuffle module, achieving only 56.1 mAP. When using channel attention and channel expansion separately, the performance improves to 57.9 mAP and 62.2 mAP, respectively. When both channel attention and channel expansion are used together, the final performance reaches 64.8 AP. These results validate the effectiveness of using channel expansion and channel attention, demonstrating significant improvements when used together compared to their individual use.

Table 5. Comparing the impact of channel expansion and channel attention in CEAM on model performance. The integration of channel expansion and channel attention demonstrates significant performance enhancement in the CEAM block.

Channel Expansion	Channel Attention	Params	GFLOPs	mAP
		3.3M	0.10	56.1
	√	3.3M	0.10	57.9
√		4.2M	0.16	62.2
√	√	4.2M	0.16	**64.8**

Number of Branches. As shown in Table 6, while maintaining a similar number of parameters and computations, the performance of the model improves as the number of branches decreases. This demonstrates that multiple branches in high-resolution networks are not essential for lightweight models. The single-branch network built by our proposed MPPose proves to be more efficient.

Table 6. Under similar parameters and GFLOPs, compare the impact of different numbers of branches on model performance.

Number of Branches	Params	GFLOPs	mAP
4	3.5M	0.17	62.1
3	3.7M	0.18	63.5
2	4.3M	0.17	64.5
1	4.2M	0.16	**64.8**

5 Conclusion

To enhance the computational efficiency and accuracy of lightweight human pose estimation models, we reconsider the design of lightweight model structure and basic block. In this paper, we design the CEAM, an improved Shuffle block, addressing the issue of reduced channel numbers due to channel split. Structurally, we adopt the mainstream single-branch architecture while implicitly retaining high-resolution features to benefit from high-resolution representation. In addition, based on the high-resolution network structure, we also implemented same-resolution feature fusion. Experimental results demonstrate that our model achieves higher accuracy with significantly lower computational requirements compared to other models. Specifically, our model outperforms Lite-HRNet-18, achieving better performance with only 85\% of its computational cost, making it more suitable for practical applications and real-time interface.

Despite using a simple structure and basic blocks to achieve lightweight design, reducing computational demands and improving inference speed, our model's parameters still has room for further reduction. This limitation may hinder its applicability in devices with extremely limited storage capacity. This issue might stem from the simplicity of the model design. In future work, it will be necessary to optimize the model's parameter requirements while maintaining a relatively simple structure to meet the needs of a broader range of applications.

References

1. F. Casado, D. P. Losada, A. Santana-Alonso *et al.*, “Pose estimation and object tracking using 2d images,” *Procedia Manufacturing*, vol. 11, pp. 63–71, 2017.
2. M. V. Da Silva and A. N. Marana, “Human action recognition in videos based on spatio-temporal features and bag-of-poses,” *Applied Soft Computing*, vol. 95, p. 106513, 2020.
3. S. Jiang, Q. Wang, F. Cheng, Y. Qi, and Q. Liu, “A unified object counting network with object occupation prior,” IEEE Transactions on Circuits and Systems for Video Technology, 2023.
4. Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, “Cascaded pyramid network for multi-person pose estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7103–7112.
5. K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, “Mask r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
6. J. D. Alejandro Newell, Kaiyu Yang, “Stacked hourglass networks for human pose estimation,” in European Conference on Computer Vision, 2016, pp. 483–499.
7. K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5693–5703.
8. S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.
9. B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 466–481.
10. 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,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5386–5395.
11. Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7291–7299.
12. E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, “Deepercut: A deeper, stronger, and faster multi-person pose estimation model,” in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14. Springer, 2016, pp. 34–50.

13. C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: A survey,” ACM Computing Surveys, vol. 56, no. 1, pp. 1–37, 2023.
14. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. AlShamma, J. Santamar´ıa, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” Journal of big Data, vol. 8, pp. 1–74, 2021.
15. G. Lan, Y. Wu, F. Hu, and Q. Hao, “Vision-based human pose estimation via deep learning: A survey,” IEEE Transactions on Human-Machine Systems, vol. 53, no. 1, pp. 253–268, 2022.
16. V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
17. T.-Y. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
18. C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang, “Lite-hrnet: A lightweight high-resolution network,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 10 440–10 450.
19. Y. Wang, M. Li, H. Cai, W.-M. Chen, and S. Han, “Lite pose: Efficient architecture design for 2d human pose estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13 126–13 136.
20. 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 preprintarXiv:1704.04861, 2017.
21. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
22. 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,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314–1324.
23. X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
24. N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.
25. Y. Wang, R. Wang, and H. Shi, “Db-hrnet: Dual branch high-resolution network for human pose estimation,” IEEE Access, vol. 11, pp. 120 628–120 641, 2023.
26. J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
27. J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, “Gather-excite: Exploiting feature context in convolutional neural networks,” Advances in neural information processing systems, vol. 31, 2018.
28. S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
29. W. Zhang, J. Fang, X. Wang, and W. Liu, “Efficientpose: Efficient human pose estimation with neural architecture search,” Computational Visual Media, vol. 7, pp. 335–347, 2021.

30. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
31. C. Neff, A. Sheth, S. Furgurson, and H. Tabkhi, “Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation,” arXiv preprint arXiv:2007.08090, 2020.
32. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “Blazepose: On-device real-time body pose tracking,” arXiv preprint arXiv:2006.10204, 2020.
33. D. Osokin, “Real-time 2d multi-person pose estimation on cpu: Lightweight openpose,” arXiv preprint arXiv:1811.12004, 2018.
34. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer, 2015, pp. 234–241.
35. Y. Li, S. Yang, P. Liu, S. Zhang, Y. Wang, Z. Wang, W. Yang, and S.-T. Xia, “Simcc: A simple coordinate classification perspective for human pose estimation,” in European Conference on Computer Vision. Springer, 2022, pp. 89–106.
36. Y. Li, Y. Chen, X. Dai, D. Chen, M. Liu, L. Yuan, Z. Liu, L. Zhang, and N. Vasconcelos, “Micronet: Improving image recognition with extremely low flops,” in Proceedings of the IEEE/CVF International conference on computer vision, 2021, pp. 468–477.
37. J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, “Run, don’t walk: chasing higher flops for faster neural networks,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 12 021–12 031.
38. F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, “Distribution-aware coordinate representation for human pose estimation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7093–7102.
39. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.
40. M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: New benchmark and state of the art analysis,” in Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, 2014, pp. 3686–3693.
41. G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, “Towards accurate multi-person pose estimation in the wild,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4903–4911.