Lightweight HRNet: A Ligtweight Network for Bottom-Up Human Pose Estimation

Jinzhen Liao, Wenhua Cui, Ye Tao, Tianwei Shi, and Lijia Shen

Abstract—**In understanding human behaviour, computers sometimes condense their analysis of human behaviour into an analysis of the state of movement of keypoints in the human body. Thus, the technique of human pose estimation provides a convenient means for machines to recognise people's behaviour. The lightweight pose estimation network enables computers to detect human poses in real-time. This paper proposes Lightweight HRNet, a bottom-up lightweight network for multi-person human pose estimation. The network is proposed based on the HRNet architecture and includes four network branches with different resolutions and two network stages. The network backbone uses the ShuffleNet model, which allows the network to be better used on smaller devices. Notably, Lightweight HRNet focuses on a problem with multi-resolution, multi-branch parallel networks: not all stages of the network contain information about the feature maps of all its branches. Therefore, we proposed the Channel Exchange Module (CEM), which improves the exchange of information between each stage of the network and each of its branches. Among the tasks of human pose estimation, the addition of CEM improved the network accuracy of Lightweight HRNet by 0.9% in the COCO2017 test-dev. Finally, the network was able to achieve 46.6% accuracy on the COCO2017 test-dev. This accuracy is superior in the study of bottom-up lightweight human pose estimation networks.**

Index Terms—**Human Pose Estimation, Lightweight Network, HRNet**

I. Introduction

MULTI-PERSON pose estimation techniques are a fundamental topic in the study of human behaviour through computer vision. Human behaviour recognition techniques, such as fall detection [1] and hazardous behaviour detection [2], often rely on human pose estimation. Depending on the order of recognition, multi-person human pose estimation has been categorised into top-down and bottom-up approaches. The top-down detection method initially employs object detection techniques to identify all individuals within the image, followed by a sequential estimation of the pose for each detected individual. The bottom-up approach begins by detecting the keypoints of all the individuals in the image, and then classifies and links these keypoints in order. As object detection and single-person human pose estimation tasks are performed sequentially, the detection time for top-down pose estimation increases with the number of people in the input image. This method has a slower detection speed compared to the bottom-up method, but it has higher accuracy.

Among the tasks of human pose estimation, especially the bottom-up multi-person human pose estimation task, the network is required to be able to capture global and local image information effectively. This is attributed to the presence of multiple detection targets with varying sizes in the input image. The multi-resolution network [3, 4, 5, 6] can effectively achieve this purpose and is therefore frequently used in human pose estimation. This particular network architecture is well suited for the task of detecting small targets such as keypoint detection. If the Stacked Hourglass Network [5] is the serial network among multi-resolution networks, which repeatedly learns the local and global information of the images to understand all the input "knowledge points" more thoroughly and deeply, then the HRNet series of networks is the parallel network among multi-resolution networks. It learns the local and global information of the input images separately and combines all the "knowledge points" in the output stage of the network. In summary, the HRNet series of networks [3, 4, 6] exhibits superior performance compared to the former [5]. It exhibits superior accuracy with a reduced number of parameters and FLOPs.

Since the development of human pose estimation tasks, several outstanding networks [7, 8, 9, 10] have emerged for human pose estimation. They are both able to achieve high detection accuracy in pose estimation tasks. However, although the accuracy of detection has improved, the increased network size has posed a significant challenge for most networks in achieving real-time attitude detection tasks. Consequently, numerous scholars have initiated investigations into lightweight networks for human pose estimation [11, 12, 13, 14]. Most of them are mainly used in single-people pose estimation tasks or in combination with object detection techniques for top-down human pose estimation tasks. Limited networks are available for lightweight human pose estimation using bottom-up methods.

Manuscript received September 19, 2023; revised January 24, 2024. This work was supported by Joint Fund Project of the National Natural Science Foundation of China (U1908218), the Natural Science Foundation project of Liaoning Province (2021-KF-12-06), and the Department of Education of Liaoning Province (LJKFZ20220197).

Jinzhen Liao is a Postgraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: G851792070@163.com).

Wenhua Cui is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (Corresponding author to provide phone: +86-133-0422-4928; e-mail: taibeijack@126.com).

Ye Tao is a Lecturer of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: taibeijack@163.com).

Tianwei Shi is an Associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: tianweiabbcc@163.com).

Lijia Shen is a Postgraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: slj7708@163.com).

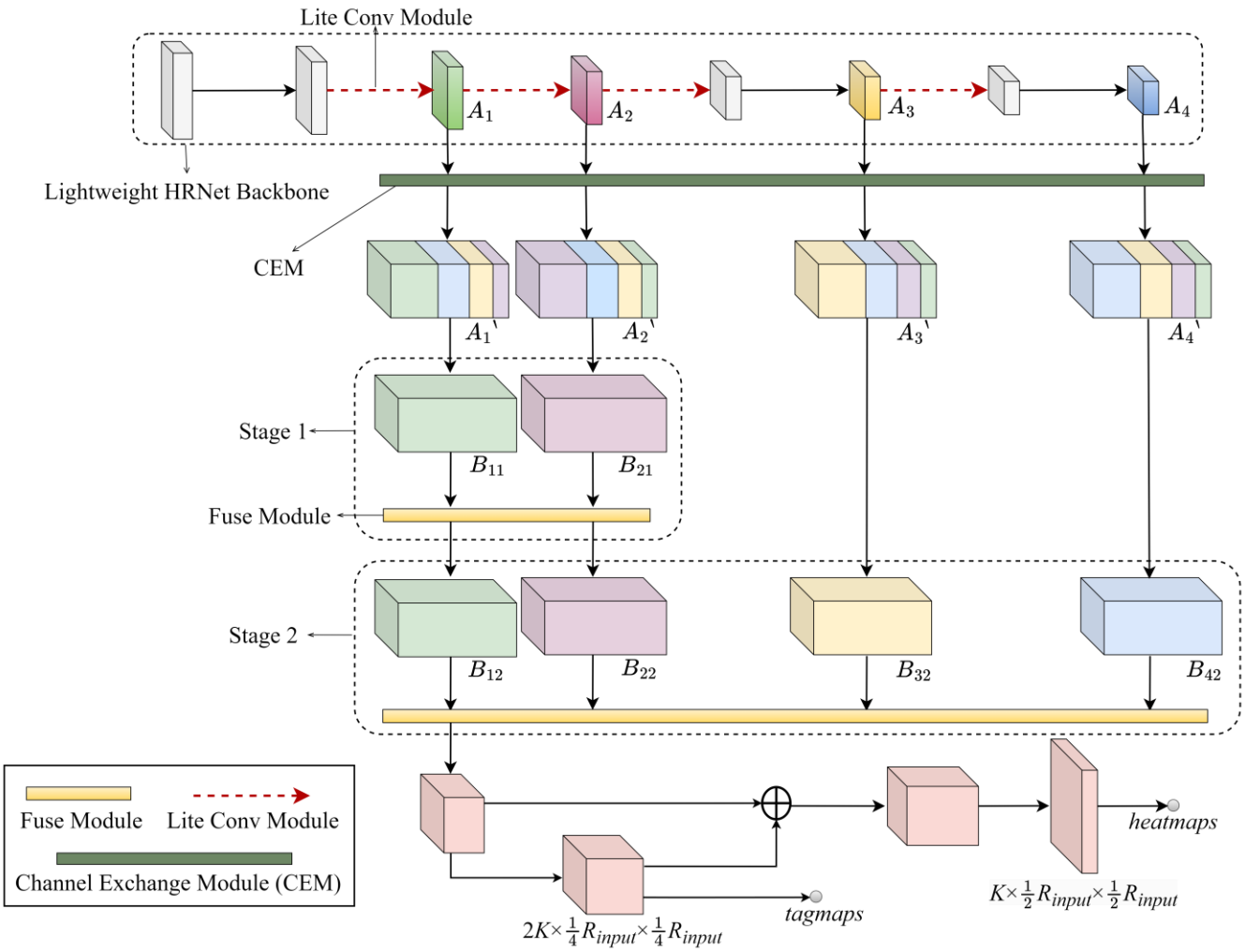

Fig. 1. Lightweight HRNet structure

Compared to the top-down approach, the bottom-up approach has a significant advantage in terms of detection speed in the multi-person human pose estimation task. This is because it eliminates the need for an additional human detector. Therefore, this paper considers it appropriate to adopt a bottom-up detection approach to achieve a lightweight design of multi-person human pose estimation network.

In this paper, we present Lightweight HRNet, a state-of-the-art lightweight network designed for bottom-up multi-person human pose estimation. Fig. 1. illustrates the structure of the network. The architectural design of a multi-resolution, multi-branch parallel network serves as the fundamental basis for Lightweight HRNet. The Lightweight HRNet adopts the parameter formulation and principles of Efficient HRNet [4]. It is inspired by the ShuffleNet series of networks [15] to propose a more efficient backbone network. The Lightweight HRNet consists of two stages and four network branches. Each stage involves a different network branch. However, the Lightweight HRNet has an issue where the information fusion between sub-networks only occurs after the convolution of each stage. As a result, there is no information exchange between branches that have not yet joined the current stage and those that have, until the stage is complete. As shown in Fig. 1, the first stage of the network does not contain information about the third and fourth branches. According to K. Su et al. [16], shuffling the channels of feature maps can improve the fusion of feature maps in different branches of multi-branch parallel network modules. Inspired by this, this paper focuses on the fact that the feature map information flow in this multi-branch parallel network is also weak, and thus proposes a more efficient Channel Exchange Module (CEM) embedded in the Lightweight HRNet. With this module embedded, information exchange between feature maps of different network branches and different network stages can also be accomplished. This ensures that each stage of a multi-resolution parallel network can contain information from other network branches, enhancing the flow of information between the feature maps of each branch between different network branches.

The Lightweight HRNet is based on the network architecture of HRNet [6]. The bottom-up human pose estimation algorithm selected is Associative Embedding [17]. The validation of Lightweight HRNet was performed in COCO2017. After the experiment, Lightweight HRNet achieved an accuracy of 46.6% in the COCO2017 test-dev, while only using 5.5M parameters and 5.5B FLOPs. The addition of the Channel Exchange Module (CEM) improves the accuracy of the network by 0.9% without any additional FLOPs or parameters in the network architecture. This good accuracy performs well and demonstrates the potential of this lightweight network for bottom-up multi-person human pose estimation.

II. Related Work

A. Single Person Human Pose Estimation

Deep learning enables computer systems to perform tasks automatically. Advances in computer vision technology have been driven by the development of deep learning, which is now being used in a number of fields [18, 19]. On the contrary, neural network research is driving the growth of the deep learning techniques. A. Toshev et al. [20] were the earliest to use neural networks for human pose estimation task. They transformed the 2D human pose estimation problem by switching the focus from an initial image processing and template matching task to a more mature approach of extracting image features and regressing keypoint positions using Convololutional Neural Networks (CNNs). There is a relatively strong similarity between some keypoints, such as between pairs of keypoints, between the wrist and elbow, and so on. To distinguish them accurately, it is important to combine the global information of the images and constantly compare similar keypoints to make correct predictions. Therefore, obtaining global information about the image is important for the quality of keypoint detection, and many studies of pose estimation networks have focused on this problem. S.-E. Wei et al. [10] proposed to improve the intermediate supervision of the network by expanding the receptive field of the convolution and incorporating the original input image throughout the network process, allowing effective learning of spatial connections between distant pixel points in the feature map. Y. Chen et al. [7] proposed a network with four feature maps of different resolutions. The higher resolution feature maps capture local information from the input image for keypoint identification, while the lower resolution feature maps provide global context to complement keypoint recognition. To address challenging keypoint recognition tasks, multiple branch networks are employed in parallel. K. Su et al. [16] proposed the idea of dimensionally exchanging feature maps of multi-branch networks, which effectively enhances the information exchange between feature maps of different branches.

B. Multi-person Human Pose Estimation

Approaches to multi-person human pose estimation have been divided into two main groups: top-down detection methods and bottom-up detection methods. The top-down approach to multi-person pose estimation is a variation of single-person pose estimation that incorporates object detection techniques. It first detects the whole person in the image and then does a single-person pose estimation for each detected individual.

The bottom-up method process is also performed in two stages. In the first stage, a method similar to top-down keypoint detection is used to extract all keypoints in the input image, and faces similar challenges to top-down keypoint detection. The detection approach used in this stage is similar to top-down keypoint detection, using the heatmap to estimate the accurate location of these keypoints. In the second stage, the detected keypoints are categorised and then grouped into the same category if they belong to the same person. This stage is a major focus of research in bottom-up pose estimation algorithms. E. Insafutdinov et al. [21] were the earliest to apply bottom-up approaches to multi-person human pose estimation. Although a human body detector was also used, the authors employed the Non-Maximum Suppression and ILP optimisation model to categorise the keypoints within the human body detection frame with overlapping relationships, which provided a new idea for the multi-person human pose estimation algorithm. Associative Embedding (AE) has been proposed by A. Newell et al. [17]. They suggest that in addition to generating a heatmap of keypoints during the keypoint detection stage, a label should be generated for each keypoint to facilitate categorization. If the corresponding label values of multiple keypoints from different categories are closely aligned, it is possible to classify them into the same category. Z. Cao et al. [22] proposed the introduction of a vector PAF as an identifier for keypoint connections between pairable human keypoints (e.g. left wrist and left elbow). Subsequently, by using the greedy algorithm, all keypoint connections can be established and matched to the different individuals.

C. Lightweight Human Pose Estimation Network

Most of the lightweight human pose estimation networks are modified from the original large-scale networks. The modification methods generally fall into two categories, one is to simplify the original network structure and the other is to introduce a network lightweighting module. Distillation learning is an effective way to recover accuracy in neural networks. M. W. Oktavian et al. [23] then applied distillation learning to disease detection, effectively improving the accuracy of the network. In the approach to simplifying the original network structure, distillation learning has also been applied to lightweight networks for pose estimation. F. Zhang et al. [13] proposed to proposed to halve the network structure of Stacked Hourglass Networks (SHN). This operation significantly reduces the network parameters. At the same time, the pruned network distilled learning from the original network, recovering a small part of the accuracy loss caused by pruning the network. D. Osokin [11] compared the efficiency of OpenPose [22] at different stages. He removed inefficient network structures, reduced the size of the network's convolution kernel in the prediction stage, and shared parameters of two different branches of the network for different tasks. C. Neff et al. [4] proposed EfficientHRNet, which sets different network parameters. The size of the resulting network structure is different depending on the parameters set, with the smallest structure having only 3.7M parameters and FLOPs of 2.1B. Bazarevsky et al. [24] employed the SHN architecture to build a tracker that accelerates human pose estimation by linking frame-to-frame poses to the body frame.

Most approaches to incorporating lightweight network modules replace the conventional convolutional modules of the original network architecture with lightweight modules to achieve parameter and FLOPs reduction goals. A. Krizhevsky et al. [25] introduced depthwise separable convolution, which successfully reduced the number of parameters of the network by splitting the ordinary convolution into two parts: depthwise convolution and pointwise convolution. X. Zhang et al. [15] proposed to shuffle the feature maps of different groups on the basis of grouped convolution, which achieves the purpose of

enhancing the information exchange between networks and ensuring the network accuracy.

Bottom-up detection is less commonly used in lightweight networks for human pose estimation because it is more accurate compared to another detection method. In this paper, we argue that bottom-up networks for human pose estimation provide inherent advantages in speed of detection. For simple pose estimation scenes, the bottom-up approach is sufficient to design a lightweight pose estimation network.

III. Lightweight HRNet Architecture

A. Introduction to the Overall Structure of the Network

Fig. 1 shows the overall architecture of Lightweight HRNet. It has a total of four network branches with different resolutions and two network stages. In the network, the Lite Conv Module of the network replaces the normal convolution in the backbone part of the network, reducing the parameters of the network to some extent. The Channel Exchange Module (CEM) of the network is embedded before the feature maps of each different resolution enter their corresponding network branches, which recovers some accuracy for the lightweight network. The Fuse Module is embedded after each stage of the network to fuse the feature map information of network branches with different resolutions to ensure the flow of information between the branches.

As shown in Fig. 1. The input image is first input to the network backbone, resulting in four feature maps of different resolutions. These feature maps are then joined together through the Channel Exchange Module (CEM) to facilitate information exchange without changing their resolution or channel. Each of these four feature maps then enters a separate branch within the network, and after each stage of processing, the branch feature maps are fused in the Fusion Module. Finally, all the feature map information generated from the branches with different resolutions is fused in the Fusion Module to produce heatmaps and tagmaps for predicted keypoints.

B. Lightweight HRNet Backbone

In the backbone of Lightweight HRNet, there are four feature maps flowing into four network branches. For convenience of description, the feature map in the *n*th branch that will enter the subnetwork in the backbone network is denoted as A_n and the feature map on the *n*th branch in the *m*th stage of the network is denoted as B_{nm}. Efficient HRNet [4] sets all the parameters according to the device and application in which it is used. In the Lightweight HRNet backbone, the dimensions of A_n and B_{nm} are set according to the method of setting network parameters in Efficient HRNet. Efficient HRNet sets the parameter *r* for users to choose the size of the network according to their needs, such as $r=-1,-2,-3,-4$. The calculation of R_{input} is shown in equation (1).

$$R_{input} = 512 + 32r \tag{1}$$

The dimension W_{a_n} of A_n is calculated as shown in equation (2).

$$W_{a_n} = g(r) \times 1.1^r \tag{2}$$

where $g(r) = 24, 40, 112, 320$ when $r=-1,-2,-3,-4$. However, in the original paper [4], when using this network for the pose estimation task, it appears as if the final result of W_{a_n} is also fine-tuned to meet the needs. The dimension $W_{b_{nm}}$ of the feature map B_{nm} on the branch is calculated as shown in equation (3).

$$W_{b_{nm}} = 32n \times 1.25^r \tag{3}$$

The size R_n of the resolution of the feature map B_{nm} on each branch is calculated according to equation (4).

$$R_n = (\frac{1}{2})^n \times R_{input} \tag{4}$$

where R_{input} is the size of the resolution of the image at the time of input to the network obtained in equation (1).

For the parameter selection of Lightweight HRNet, we experimented with the network parameters $r=0,-1,-2$. The network representation was found to be best with Lightweight HRNet when $r=-2$ was defined. Compared to Efficient HRNet, Lightweight HRNet improves accuracy by 0.8% compared to the H_{-3} network in Efficient HRNet, while maintaining basically the same FLOPs and parameters as the H_{-3} network in Efficient HRNet. Therefore, in this paper, $r=-2$ is chosen as the parameter of Lightweight HRNet for further optimisation.

According to equations (1) - (4), the convolution process and the parameters of each feature map of the backbone network of Lightweight HRNet when $r=-2$ are shown in Table I, and the sizes of its feature maps for different branches with different network stages are shown in Table II.

TABLE I
Lightweight HRNet Backbone ($r=-2$)

Feature map serial number	Convolutional operations in backbone
Input	$3\times448\times448$
	$\text{conv}=3\times3, \text{stride}=2, \text{out}_{\text{channel}}=16$
1	$16\times448\times448$
	Lite Conv Module
2	$24\times112\times112$
	Lite Conv Module
3	$32\times56\times56$
	Lite Conv Module
4	$64\times28\times28$
	$\text{conv}=3\times3, \text{stride}=2, \text{out}_{\text{channel}}=96$
5	$96\times28\times28$
	Lite Conv Module
6	$160\times14\times14$
	$\text{conv}=3\times3, \text{stride}=2, \text{out}_{\text{channel}}=264$
7	$264\times14\times14$

C. Lite Conv Module

The Lite Conv Module of the network is shown in Fig. 2, which is inserted in the middle of the partial convolution

TABLE II
LIGHTWEIGHT HRNET BRANCH FEATURE MAP SIZE ($r = -2$)

	Branch1	Branch2	Branch3	Branch4
Backbone	24×112×112	32×56×56	96×28×28	264×14×14
Afer CEM	24×112×112	32×56×56	96×28×28	264×14×14
Stage1	21×112×112	42×56×56		
Stage2	21×112×112	42×56×56	83×28×28	166×14×14

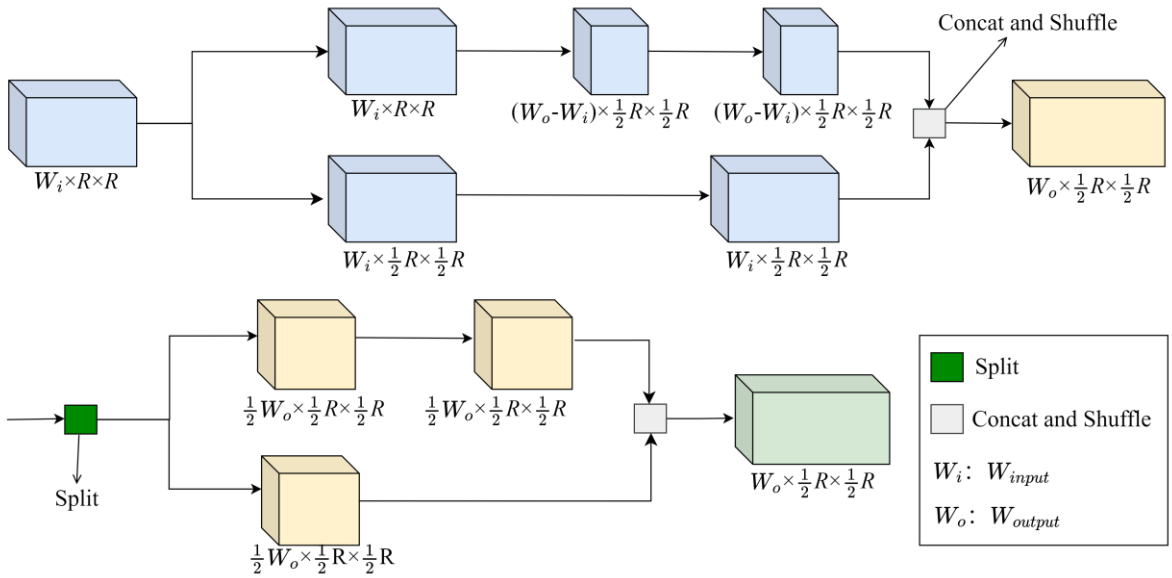

Fig. 2. Lite Conv Module structure

process of the backbone network. This module follows the idea of group convolution [25] and ShuffleNet [15] by replacing part of the normal convolution process with a two-stage convolution operation.

In the previous section, we determined the dimensions of each feature map in the backbone of the network, and it becomes clear that the entire convolution process consists of a sequence of operations aimed at increasing dimensionality. The first stage of the convolution operation in the Lite Conv module involves unifying the dimensions of the input feature maps to define the size, followed by the simultaneous feeding of these feature maps into both branches. For ease of description, the two branches of the network in the first stage will be named C_{11} and C_{12}, respectively. Assume that the dimension size of the input feature map of a module identified in the previous section is W_{input} and the dimension size of the output feature map is W_{output}. In the C_{11} branch, the size of the feature map dimension is not changed, so the final feature map output in this branch has dimension W_{output}. However, the operation of dimensionality reduction of the original feature map is required in C_{12}. The size of dimensionality reduction will be determined by the dimension of the final output feature map and the dimension of the input feature map, i.e., the dimension of the final output feature map in the C_{12} branch of the network is $W_{output} - W_{input}$. The final feature map output in this stage is the stitching of the feature maps output from the C_{11} and C_{12} networks, and the size of the output feature map dimension is still W_{output}. This is equivalent to the fact that the convolution operation of C_{12} is the addition of the dimensions of the input feature maps to the dimensions of the output feature maps, so that the dimensional size of the feature maps after the stitching of the output feature maps of C_{11} and C_{12} matches the dimensional size of the output feature maps.

The second stage of the convolution operation is added to ensure the depth of the network. In the second stage of the convolution operation, the dimensional size of the feature map is no longer altered and the feature map is divided equally into two parts. The network performs a convolution operation on only one part of it, leaving the other part of the feature map without any processing. Subsequently, the feature maps of the two parts were stitched together. It is worth noting that in both the first and second stages of the convolution operation, the shuffle module is added to improve the information exchange between the feature maps, that is, the feature maps are disrupted and reorganised in terms of different dimensions. Instead of two standard convolution operations, the network uses the Lite Conv Module, which is able to effectively reduce the number of parameters in the network while maintaining the depth of the network compared to normal convolution.

D. Channel Exchange Module

The whole process of the Channel Exchange Module (CEM) is shown in Fig. 3. The operation process of the

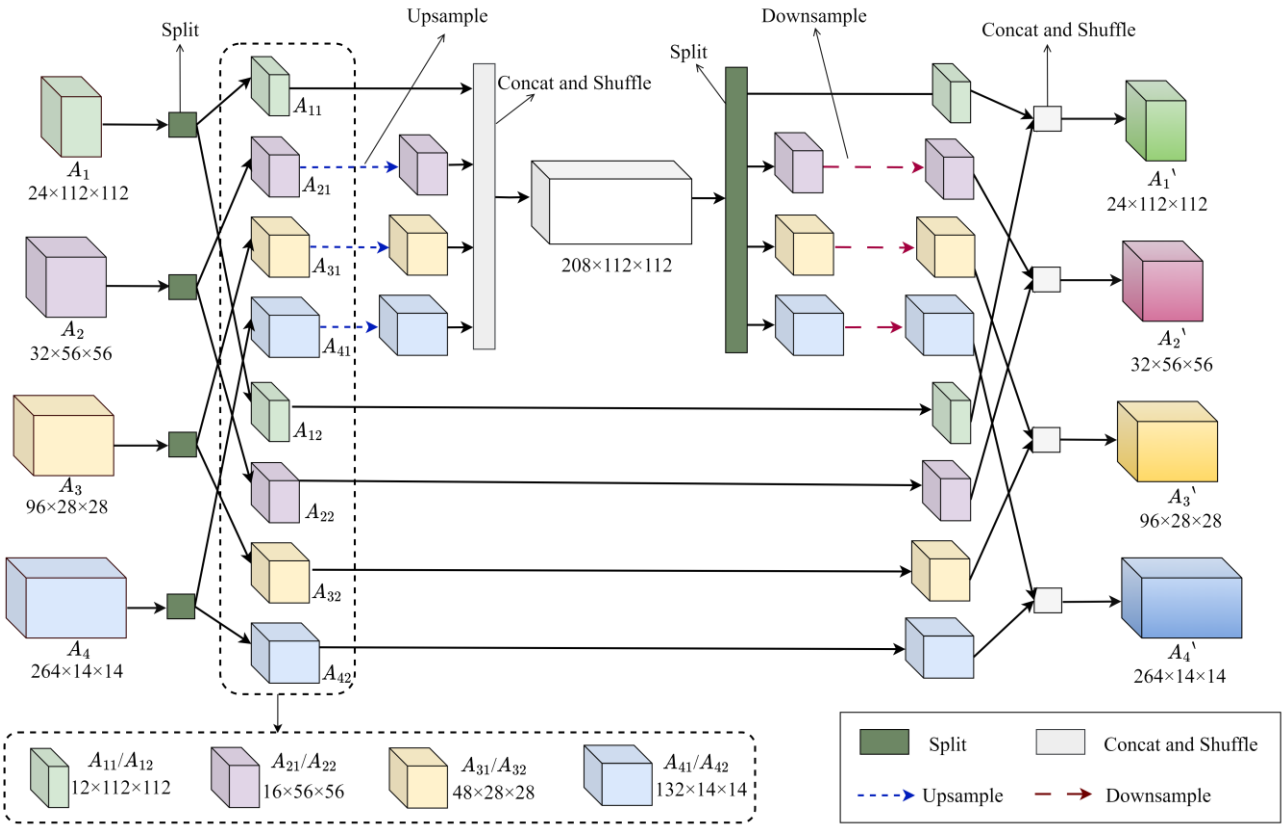

Fig. 3. Channel Exchange Module structure

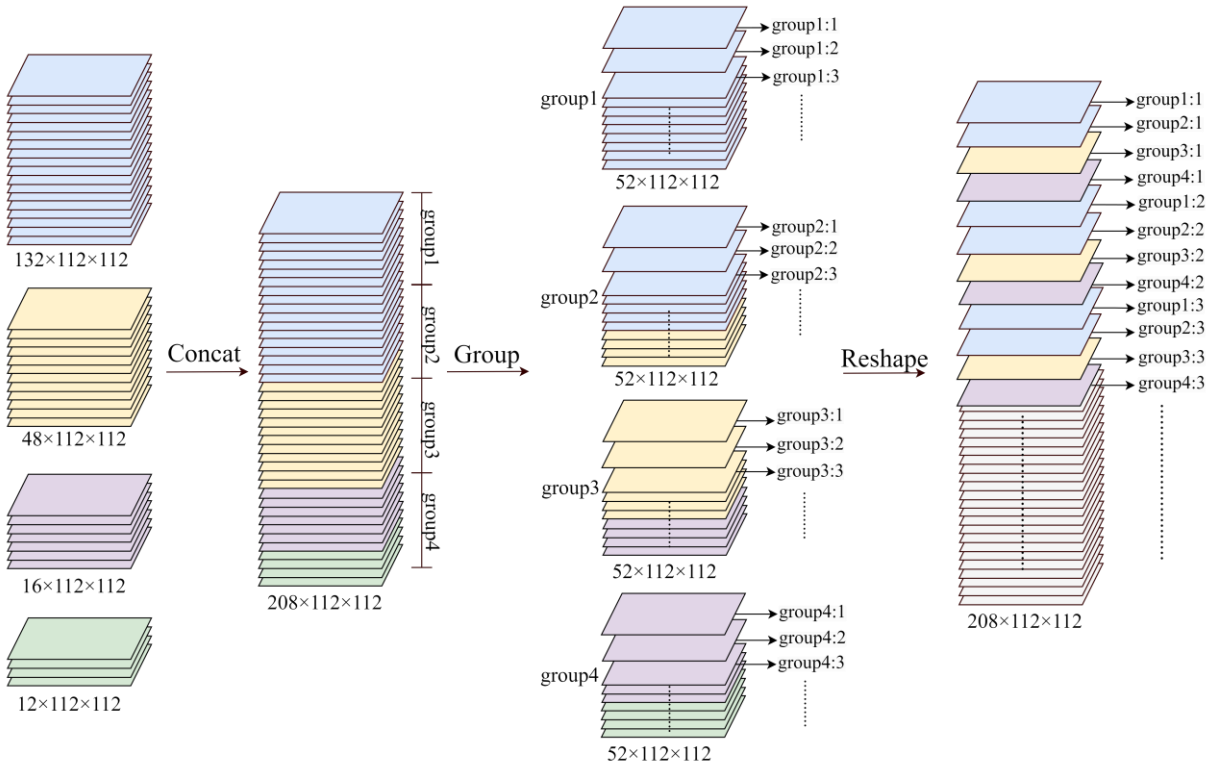

Fig. 4. The operation process of the concat and shuffle module in CEM [15]

concat and shuffle module in CEM is shown in Fig. 4 [15]. It should be noted that the stitching of feature maps requires that the feature maps have the same resolution size, but in the Lightweight HRNet network these feature maps do not have the same resolution size. To address this problem, CEM unifies all their resolutions by the operation of up-sampling into the size of the feature map $R \times R$ which is the largest resolution among them, and this does not increase any FLOPs and parameters amount of the network. After the task of channel exchange is completed, the CEM will use normal convolution with a convolution kernel size of 3×3 and a step size of $2n$ to complete the downsampling operation of the feature map to recover the size of the feature map resolution. Since the network has four network branches, the

whole CEM is equivalent to four convolution operations, just enough to replace the dimensionality-decreasing convolution performed by A_n when it enters the branch network. In this way, the CEM does not add any parameters or FLOPs to the whole network.

For the ease of description, the feature maps after CEM are named as A_1', A_2', A_3', A_4'. Although the network needs to obtain more complete information about the input image by enhancing the exchange of information between feature maps, the network still expects that after the channel exchange is completed, A_n' can still retain most of its own dimensional information A_n, rather than having its own dimensional information completely replaced. Moreover, in the process of channel exchange, each feature map also expects to exchange as much as possible with the dimensions of feature maps that have less relevance to itself. Therefore, CEM divides A_n into two equal parts, takes only one part to exchange information with other feature maps, and puts the information of the other part into A_n' as it is.

After the above channel exchange in CEM, the number of dimensions of the original feature map A_n contained in A_n' is shown in Table III-IV. Table III shows the case when all the feature maps of the four branches are performed with channel exchange. Table IV shows the case when only half of the four branching feature maps are channel exchanged. In the first stage of Lightweight HRNet, the feature map only has the information of the first and second branch feature maps, but lacks the information of the third and fourth branch feature maps, then it is expected that A_1' and A_2' can be exchanged to the dimension information of A_3 and A_4 as much as possible in the CEM. However, it is worth noting that these feature maps have different and widely varying dimensions, and if all the dimensional information of the feature maps is exchanged according to the allocation shown in Fig. 4, it may result in most or all of the dimensional information of the feature maps themselves being exchanged. As shown in Table III, if all the information of the feature map is exchanged dimensionally, the information of A_3 in the feature map after exchanging channel information A_3' only accounts for 1/4 of its information, and even A_2' doesn't contain the information of A_2. It doesn't make sense for a feature map to lose most or all of its own information as it goes about exchanging information.

TABLE III
THE EXCHANGE OF FEATURE MAPS OF EACH BRANCH BEFORE JOINING CEM

	A_1 (24)	A_2 (32)	A_3 (96)	A_4 (264)
A_1' (24)	6	8	10	0
A_2' (32)	0	0	14	18
A_3' (96)	6	8	24	58
A_4' (264)	12	16	48	188

TABLE IV
THE EXCHANGE OF FEATURE MAPS OF EACH BRANCH AFTER JOINING CEM

	A_1 (24)	A_2 (32)	A_3 (96)	A_4 (264)
A_1' (24)	12	0	12	0
A_2' (32)	0	20	0	12
A_3' (96)	3	4	60	29
A_4' (264)	9	8	24	223

Therefore, as shown in Fig. 3, each feature map A_n is segmented in the CEM. It is divided equally into two parts, A_{n1} and A_{n2}, with only one half of A_{n1} exchanging dimensions, and the other half, A_{n2}, being left unprocessed. Once the dimensions of A_{n1} have been exchanged, they are stitched together with the corresponding A_{n2} which has not been processed in any way. As shown in Table IV when only half of the feature map is taken for information exchange. Such approach allows each feature map A_n to keep at least half of its own information after the channel exchange is completed, so that no more information will be lost when the resolution size of A_n' is recovered later. In addition, when assigning the dimensions of the features again, CEM adjusts the order of the feature maps when assigning the dimensions, so that A_1', A_2' can contain as much information as possible about the dimensions in A_3, and A_4.

IV. EXPERIMENTS

Lightweight HRNet experiments were performed in Python 3.9, pytorch 1.13.0 and all training was done on the NVIDIA GeForce RTX 3090 device. The experiment is described and analysed below.

A. Experimental Dataset

The dataset used for Lightweight HRNet is COCO2017. The COCO dataset [26] is a large dataset that can be used for various tasks, including object detection, semantic segmentation, and pose detection. It contains 250,000 pedestrians with keypoint annotations, each with 17 keypoints, which can be used for the human pose estimation task of the network. The COCO2017 dataset is divided into three sets: training, validation, and test. The training set, which is used for training the network, contains more than 118,000 images, the validation set contains more than 5,000 images, and the test set contains more than 40,000 images. The accuracy of human pose estimation on this dataset has been validated through widely recognized methods. The COCO dataset has been used to train many pose estimation tasks, and the resulting models have been validated on both the COCO test-dev and COCO val datasets. The experimental results of Lightweight HRNet are validated on COCO2017 test-dev and COCO2017 val to enable comparison with other attitude networks.

B. Evaluation Indicators

The evaluation indicators of the COCO dataset in human pose estimation mainly refer to its evaluation indicators in

object detection: Average Precision (AP) and Average Recall (AR). In our task, Ground Truth (GT) denotes the true position of human keypoints manually labelled in the dataset, and Bounding Box (Bbox) denotes the position of human keypoints predicted by the network, and the similarity measure between them is OKS. The full name of OKS is Object Keypoint Similarity and it is calculated as shown in equation (5).

$$OKS_p = \frac{\sum_i \exp(-d_{p^i}^2 / 2S_p^2\sigma_i^2)\delta(v_{p^i} > 0)}{\sum_i \delta(v_{p^i} > 0)} \tag{5}$$

where p denotes a person among the GT. p^i represents the ith keypoint of the person, and d_{p^i} denotes the euclidean distance between the ith detected keypoint of the person in the Bbox and the ith keypoint of the person in the GT. It is calculated as shown in equation (6).

$$d_{p^i} = \sqrt{(x_i' - x_{p^i})(y_i' - y_{p^i})} \tag{6}$$

where (x_i, y_i) is Bbox, i.e., the position of the ith person's keypoint detected by the network; (x_{p^i}, y_{p^i}) is GT, i.e., the position of the ith keypoint of the pedestrian p manually labelled. v_{p^i} is a manually labelled parameter, and $v_{p^i} = 0$ means that this keypoint is not labelled, then no computation is done for this keypoint either. S_p denotes the scale factor of the person p among the labelled pedestrians, which is the area of the pedestrian detection boxes in the image. σ_i denotes the difficulty of detection of the corresponding keypoint on the network.

The human pose estimation task mainly takes the average accuracy AP as an evaluation indicator. AP is computed differently for top-down and bottom-up method. This section only concerns the AP settings for bottom-up human pose estimation. Assuming that there are M individuals labelled among the input images and the number of people predicted by the network is N. Since the correspondence between M and N is not known, it is necessary to sequentially calculate the OKS of the labelled M individuals respectively with the predicted N. Then a threshold T is artificially determined as the criterion for the network to correctly detect the human body pose maps. The bottom-up AP values are calculated as shown in (7).

$$AP = \frac{\sum_m \sum_p \delta(OKS_p > T)}{\sum_m \sum_p 1} \tag{7}$$

where m represents the human pose map in GT and p represents the human pose map detected by the network. The mAP is a commonly used detection indicator. Specifically, it is to obtain multiple AP values by setting different artificial thresholds T in the AP indicator, and mAP can be obtained by averaging these AP values. In this paper, mAP is used as an evaluation indicator for the pose estimation.

In addition to this, since we are doing research on lightweight networks, this paper also uses the number of parameters and Floating Point Operations (FLOPs) as evaluation indicators.

C. Experimental Results

1) Validation results of Lightweight HRNet at different parameters before adding the CEM

The validation results of the network on COCO2017 val

TABLE V
COMPARISON OF NETWORK EFFECT BETWEEN LIGHTWEIGHT HRNET AND EFFICIENT HRNET IN DIFFERENT PARAMETERS

Method	Network	Backbone	Input size	#Params	FLOPs	AP(val)	AP(test-dev)
Efficient HRNet	H_0	B_0	512	23.3M	25.6B	64.8	64.0
	H_{-1}	B_{-1}	480	16M	14.2B	59.2	59.1
	H_{-2}	B_{-2}	448	10.3M	7.7B	52.9	52.8
	H_{-3}	B_{-3}	416	6.9M	4.2B	44.8	44.5
	H_{-4}	B_{-4}	384	3.7M	2.1B	35.7	35.5
Ours (Lightweight HRNet)	$r = 0$	-	512	13.2M	15.6B	55.2	54.9
	$r = -1$	-	480	8.8M	9.3B	49.0	48.5
	$r = -2$	-	448	5.9M	5.5B	45.4	45.7

Fig. 5. Lightweight HRNet results comparison before and after CEM is added

and COCO2017 dev-test for the selection of the parameter r before adding the CEM to Lightweight HRNet are shown in Table V. At $r=0,-1$, the Lightweight HRNet does not perform well compared to the Effcient HRNet. Taking the validation results in COCO2017 dev-test as an example, at $r=0$, the number of parameters and the FLOPs amount of Lightweight HRNet are above and below H_{-1}, but the accuracy is below H_{-1}; when $r=-1$, the number of parameters and the FLOPs of Lightweight HRNet are around H_{-2}, but the accuracy is below H_{-2}. However, for $r=-2$, the number of parameters and the FLOPs of Lightweight HRNet can be reduced by 42.7% and 28.5% respectively compared to H_{-2}, and its parameters and the FLOPs are in the range of H_{-3}, while the accuracy is 1.2% higher than that of H_{-3}.

2) Comparison of results before and after adding CEM

The validation results of Lightweight HRNet (LH) before and after the addition of the CEM on the COCO2017 val and COCO2017 dev-test datasets are shown in Table VI. With the addition of CEM, the Lightweight HRNet network was able to improve accuracy by 1.2% on the COCO2017 val dataset and 0.9% on the COCO2017 dev-test dataset. And after comparison, it is found that after adding CEM to the network, the network is more accurate in capturing the keypoints in the images, as shown in Fig. 5.

TABLE VI
COMPARISON OF LIGHTWEIGHT HRNET BEFORE AND AFTER THE ADDITION OF CEM

Method	#Params	FLOPs	AP(val)	AP(test-dev)
LH	5.9M	5.5B	45.4	45.7
LH+CEM	5.9M	5.5B	46.6	46.6

3) Comparison with other bottom-up lightweight networks for human pose estimation

The validation results of Lightweight HRNet with other bottom-up lightweight networks at COCO2017 val are shown in Table VII. Compared to Lightweight OpenPose, Lightweight HRNet is 39% less FLOPs and 43% more parameters than Lightweight OpenPose in terms of computational complexity; In detection accuracy, Lightweight HRNet's AP improved by 3.8%. Compared with Efficient HRNet's H_{-3}, which has similar number of parameters and FLOPs, Lightweight HRNet's FLOPs increases by 31% and the parameters decreases by 14%, but Lightweight HRNet's AP improves by 1.8. Compared to the smaller model H_{-4}, Lightweight HRNet's model parameters and FLOPs are both increased, but the accuracy is improved by 10.9%.

TABLE VII
COMPARISON OF THE EFFECTIVENESS OF LIGHTWEIGHT HRNET WITH OTHER BOTTOM-UP LIGHTWEIGHT NETWORKS

Method	Input size	#Params	FLOPs	AP
H_{-3}	416	6.9M	4.2B	44.8
H_{-4}	384	3.7M	2.1B	35.7
Lightweight OpenPose	368	4.1M	9.0B	42.8
Ours	448	5.9M	5.5B	46.6

4) Comparison with other bottom-up human pose estimation networks

The validation results of Lightweight HRNet with other bottom-up human pose estimation networks on COCO2017 test-dev are shown in Table VIII. Compared to other state-of-the-art bottom-up networks, Lightweight HRNet has one-fifth or even one-tenth the number of parameters and FLOPs of other large-scale pose estimation networks (e.g. Hourglass [5], HigherHRNet [3]), although it has no advantage in detection accuracy in the pose estimation task. If Lightweight HRNet is used in some edge devices, this accuracy is enought for human pose estimation tasks in simple scenarios.

V. CONCLUSION

In this paper, we propose Lightweight HRNet, a new bottom-up lightweight human pose estimation network. The

TABLE VIII
COMPARISON OF LIGHTWEIGHT HRNET WITH OTHER BOTTOM-UP HUMAN POSE ESTIMATION NETWORKS

Method	Network	Backbone	Input size	#Params	FLOPs	AP
Associative Embedding	Hourglass	Hourglass	512	277.8M	206.9B	56.6
	HRNet	HRNet-W32	512	28.5M	38.9B	64.1
	HigherHRNet	HRNet-W32	512	28.6M	47.9B	66.4
	HigherHRNet	HRNet-W48	640	63.8M	154.3B	68.4
Efficient HRNet + Associative Embedding	H_0	B_0	512	23.3M	25.6B	64.0
	H_{-1}	B_{-1}	480	16M	14.2B	59.1
	H_{-2}	B_{-2}	448	10.3M	7.7B	52.8
	H_{-3}	B_{-3}	416	6.9M	4.2B	44.5
	H_{-4}	B_{-4}	384	3.7M	2.1B	35.5
PifPaf	-	ResNet-512	-	-	-	66.7
PersonLab	-	ResNet-512	1401	68.7M	405.5B	66.5
OpenPose	-	-	-	25.94M	160B	61.8
Ours	-	-	448	5.9M	5.5B	46.6

advantage of bottom-up human pose estimation is that it does not require additional human target detection tasks and can detect human pose in one step. This detection is ideal for real-time detection tasks. Lightweight HRNet is a lightweight network designed around the HRNet architecture with two network stages and four network branches. Lightweight HRNet backbone consists of normal convolution and Lite Conv Module, which compresses the number of parameters and FLOPs of the model compared to normal convolution. The parameters of the network backbone were determined on the calculation of the parameters of Efficient HRNet. At the same time, the network adds the CEM, which is placed before the backbone network's feature map enters the branch network, effectively improving the exchange of information between different branches of the network.

In the COCO2017 dev-test dataset, the addition of CEM effectively brought a 0.9% accuracy improvement to the network. Compared to other top-down human pose estimation networks, Lightweight HRNet has only 5.9M parameters and 5.5B FLOPs. It achieves an accuracy of 46.6% on COCO2017 test-dev, which is outperforming the lightweight bottom-up human pose estimation network. The low complexity of Lightweight HRNet enables it to be flexibly applied to a variety of lightweight devices for real-time human pose estimation tasks.

REFERENCES

[1] H. Zhang, W. Cui, T. Shi, Y. Tao and J. Zhang, "ATMLP： Attention and Time Series MLP for Fall Detection," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 1, pp. 58-65, 2023.

[2] K. A. Shahrim, A. H. A. Rahman and S. Goudarzi, "Hazardous Human Activity Recognition in Hospital Environment Using Deep Learning," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 3, pp. 748-753, 2022.

[3] 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. 5386-5395, 2020.

[4] C. Neff, A. Sheth, S. Furgurson and H. Tabkhi, "Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation," arXiv: 2007.08090, 2020.

[5] A. Newell, K. Yang and J. Deng, "Stacked hourglass networks for human pose estimation," *Computer Vision–ECCV 2016: 14th European Conference*, pp. 483-499, 2016.

[6] K. Sun, B. Xiao, D. Liu and J. Wang, "Deep high-resolution representation learning for human pose estimation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693-5703, 2019.

[7] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu and J. Sun, "Cascaded pyramid network for multi-person pose estimation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7103-7112, 2018.

[8] M. Kocabas, S. Karagoz and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 417-433, 2018.

[9] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler and K. Murphy, "Towards accurate multi-person pose estimation in the wild," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4903-4911, 2017.

[10] S.-E. Wei, V. Ramakrishna, T. Kanade and Y. Sheikh, "Convolutional pose machines," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724-4732, 2016.

[11] D. Osokin, "Real-time 2d multi-person pose estimation on cpu: Lightweight openpose," arXiv: 1811.12004, 2018.

[12] 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. 10440-10450, 2021.

[13] F. Zhang, X. Zhu and M. Ye, "Fast human pose estimation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3517-3526, 2019.

[14] Z. Zhang, J. Tang and G. Wu, "Simple and lightweight human pose estimation," arXiv: 1911.10346, 2019.

[15] X. Zhang, X. Zhou, M. Lin and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848-6856, 2018.

[16] K. Su, D. Yu, Z. Xu, X. Geng and C. Wang, "Multi-person pose estimation with enhanced channel-wise and spatial information," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5674-5682, 2019.

[17] A. Newell, Z. Huang and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *In Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2274-2284, 2017.

[18] H. Begum, M. M. Islam, H. S. Eva, N. H. Emon and F. A. Siddique, "Deep Learning Networks for Handwritten Bangla Character Recognition," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 4, pp. 1170-1182, 2023.

[19] X. Sun, W. Cui, Y. Tao and Z. Wang, "Flame Image Detection Algorithm Based on Computer Vision," *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1142-1158, 2023.

[20] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653-1660, 2014.

[21] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4929-4937, 2016.

[22] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. pp. 172-186, 2018.

[23] M. W. Oktavian, N. Yudistira and A. Ridok, "Classification of Alzheimer's Disease Using the Convolutional Neural Network (CNN) with Transfer Learning and Weighted Loss," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 947-953, 2023.

[24] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang and M. Grundmann, "Blazepose: On-device real-time body pose tracking," arXiv: 2006.10204, 2020.

[25] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. pp. 2012.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft coco: Common objects in context," *Computer Vision–ECCV 2014: 13th European Conference*, pp. 740-755, 2014.