IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 2, 2021, accepted March 23, 2021, date of publication March 26, 2021, date of current version April 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069102

Fast and Lightweight Human Pose Estimation

HAOPAN REN, WENMING WANG, KAIXIANG ZHANG, DEJIAN WEI, YANYAN GAO, AND YUE SUN
School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
Corresponding author: Wenming Wang (wenmingwang2004@aliyun.com)

This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1406302.

ABSTRACT Although achieving significant improvement on pose estimation, the major drawback is that most top-performing methods tend to adopt complex architecture and spend large computational cost to achieve higher performance. Due to the edge device's limited resources, its top-performing methods are hard to maintain fast inference speed in practice. To address this issue, we proposed the fast and lightweight human pose estimation method to maintain high performance and bear the less computational cost. Especially, the proposed method consists of two parts, i.e., the fast and lightweight pose network (FLPN) for pose estimation and a novel lightweight bottleneck block for reducing computational cost, which can integrate the simple network and lightweight bottleneck into an efficient method for accurate pose estimation. In terms of lightweight bottleneck block, we introduce the structural similarity measurement (SSIM) to refine the appropriate ratio of intrinsic feature maps and reduce the model size. Furthermore, an attention mechanism is also adopted in our lightweight bottleneck block for modeling the contextual information. We demonstrate the performance of the proposed method with extensive experiments on the two standard benchmark datasets by comparing our method with state-of-the-art methods. On the COCO keypoint detection dataset, our proposed method attains a similar accuracy with these state-of-the-art methods, but the computational cost of these top-performing methods is more than 7 times that of ours.

INDEX TERMS Human pose estimation, structural similarity, cheap operation, lightweight block.

I. INTRODUCTION

The goal of estimating human pose based on input images can be simplified to precisely localize human anatomical keypoints (elbows, wrists, knees, etc.). Human pose estimation which is a fundamental task in computer vision is extensively adopted for action recognition [24], [25], pose tracking [26], and human-computer interaction [27].

Recently, multiple tasks related to human pose estimation have been extensively studied in various fields [28]–[30], [33]. We pay attention to single-person pose estimation, which is the basis of relevant vision tasks, such as multi-person pose estimation, video-based pose estimation, and pose tracking.

Similar as plenty of vision tasks, great advances on human pose estimation have been achieved by deep convolutional neural networks (DCNNs) [10], [12], [13], [15], [18], [19], [24], [25], [29]–[33]. Through the pioneering work in [20], [31], the performance on the two baseline benchmarks has reached saturation in the past two years. Particularly, the accuracy on the MPII benchmark [14] has been promoted from roughly 80% PCKH@0.5 to higher than 90% [10], [12], [13]. For the challenging COCO human pose benchmark [11], the mAP score is significantly increased from 60.5 (Openpose [19]) to 77.0 (HRNet-w48 + extra data [12]) in recent three years. Accompanied by the quick development of human pose estimation, the desire for lightweight and quick inference speed pose estimation method has been proposed.

We contend that applying a lightweight model for real-time human pose estimation is one of the major unaddressed issues. To the best of our knowledge, there have been a quite few works on the lightweight of human pose estimation methods. However, the lightweight human pose estimation networks, with small model size, light computation consuming, and high accuracy are suitable to directly deploy on resource-limited devices such as mobile phones and smart laptops. Majority of state-of-the-art methods which reach higher performing level are always related to complex networks, with mass parameters and numerous float-point

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Sanchez.

operations (FLOPs). Despite their top performance, the delay based on inference time is one of the major drawbacks for these complex models with large computation. Besides, the demand for high memory is indeed for complex models with huge amounts of parameters.

Intuitively, if we aim at designing lightweight pose estimation networks, it is reasonable to focus on simple pose estimation networks and efficient bottleneck blocks. Among top-performing networks, SimpleBaseline [13] has provided prior knowledge on designing a simple yet efficient network for pose estimation and exploring how simple could an efficient model be. Inspired by their graceful design, the lightweight bottleneck of Lightweight Pose Network (LPN) [10] is proposed to exploit the best of choice depth-wise convolution for the low memory demanding network architecture. At the same time, many lightweight bottleneck blocks adopted for image classification are put forward to replace these standard bottleneck blocks such as mobilenet-v3 bottleneck [6] and ghostnet bottleneck [9]. Practically, these methods can significantly reduce the model size and computational complexity without too much performance degradation. To design an efficient lightweight network for human pose estimation, we need to explore the best balance between accuracy and computational cost.

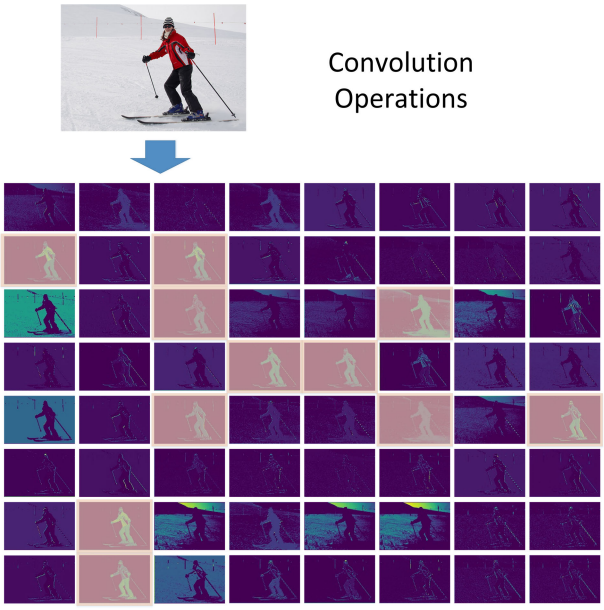


FIGURE 1. The problem of massive redundancy feature maps. In the process of convolutional operations, intermediate feature maps often contain comprehensive redundancy and result in extensive computational cost. It is important to develop an efficient method to estimate human pose with low computational cost and less redundancy.

The major difficulty lies in how to trade off the performance and lightweight size of the network. We address this problem by using a simple network with a novel lightweight bottleneck. As is shown in figure 1, the method of SSIM is introduced to compare similarity among feature maps and determine the ratio of intrinsic feature maps. A novel bottleneck block is proposed to reduce computational cost and maintain efficient performance. (The method is described in greater detail in the following Section III)

To demonstrate the effectiveness and efficiency of the proposed fast and lightweight human pose estimation method, extensive experiments were conducted to prove the superior performance over two benchmark datasets: the COCO keypoint detection dataset [11] and the MPII Human Pose dataset [14]. The experimental results confirm that our proposed method has an extremely small model size and computational complexity than these existing state-of-the-art methods [12], [13].

The contributions of this paper are as follows.

- After observing most of these state-of-the-art methods adopt standard bottlenecks in their network with heavy computational cost, we design a novel bottleneck for drastically reducing the parameters and floating-point operations (FLOPs). This allows us to deploy complex architecture network on limited resources computational platform.
- We propose a lightweight human pose estimation method by redesigning a quite simple network with surprising effectiveness. Further, the series of bottlenecks with lightweight designing are complementarily trained following a beginning block with two convolutional layers to study the high-to-low resolution representation for predicting accurate heatmaps.

The remainder of this paper is organized as follows: Section II describes the related works on top-performing and lightweight human pose estimation networks, lightweight block for various vision tasks and attention mechanism. Section III illustrates the proposed lightweight bottleneck and simple network. The detailed implementations and experiment results are presented in Sections IV and V, respectively. Ultimately, Section VI summarizes the paper.

II. RELATED WORKS

A. TOP-PERFORMING HUMAN POSE ESTIMATION

With the introduction of DeepPose by Toshev and Szegedy [20], the problem of human pose estimation has transformed from a pictorial stricture to a DNN-based keypoints regression. Since then, a mass of studies in the human pose estimation field have achieved significant improvements by adopting DCNNs [10], [12], [13], [15], [18]–[20], [24]–[27]. There are two mainstream approaches, keypoints regression [20] and keypoints heatmap [10], [12], [13], which have become dominant in this field. Compared with the method of keypoints regression, keypoints heatmap is extensively adopted in human pose estimation tasks with an overwhelming superiority in the quality of performance.

Newell *et al.* [32] proposed a dominant approach called Stacked Hourglass Network on the MPII benchmark [14], which is widely adopted by superior methods. Its features are processed in a multi-stage architecture with repeated Bottom-up, Top-down processing and skip layer connection are critical to capture the various spatial relationships between body parts. Chen *et al.* [34] proposed a method called the Cascaded Pyramid Network (CPN) which integrates

all levels of feature representations to relieve the problem of these invisible keypoints. To obtain rich high-resolution representations for accurate and precise human pose estimation, Sun *et al.* [12] proposed a high-resolution representation (HRNet) that achieved state-of-the-art performance by connecting the multi-resolution subnetworks in parallel. HRNet starts with a high-resolution subnetwork as the first stage and gradually adds high-to-low resolution subnetworks one by one to form more stages, and connects the multi-resolution subnetworks in parallel. Repeatedly performing multi-scale fusions among these parallel multi-resolutions subnetworks, HRNet can obtain rich high-to-low resolution representations from other parallel representations over and over and finally get the rich high-resolution representations. Through its superior pose estimation results over two benchmark datasets, Sun *et al.* empirically demonstrated the effectiveness of the multi-resolution subnetworks and the repeated multi-scale fusions.

Most of these prior works mainly focus on how to design a top-performing pose estimation method by adopting complex architecture or expensive computation cost model, ignoring these limitations on edge devices such as time-consuming and high memory demanding.

B. LIGHTWEIGHT POSE ESTIMATION

Lightweight design for pose estimation has attracted little attention from outstanding researchers. Recently, there rarely exists research on lightweight design to improve the efficiency of pose estimation. For model compression and execution speedup, Bulat and Tzimiropoulos [35] binarized the network architecture to adapt for edge devices. However, it suffers from a performance drop by a large margin. After observing these complex top-performing pose estimation algorithms, Xiao *et al.* [13] proposed the Simple Baseline method that is based on a residual backbone network followed by several deconvolutional layers. It provides us a new idea that simple network architecture can also achieve excellent performance on the COCO2017 benchmark [11]. Inspired by the design principles of Simple Baseline [13], Zhang *et al.* [10] provided a lightweight pose network (LPN) that has obvious superiority in terms of model size, computational complexity, and inference speed. Further, Zhang *et al.* empirically demonstrated the efficiency and effectiveness of their lightweight network on the challenging COCO2017 benchmark [11].

C. LIGHTWEIGHT BLOCK

The depth of representations is of crucial significance for the human pose estimation task. Based on comprehensive empirical evidence, He *et al.* [23] experimentally demonstrated that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. Therefore, most of these aforementioned methods adopt the ResNet series as their backbone network that is substantially deeper than those used previously. However, top-performing methods based on ResNet are not suitable to directly deploy on resource-limited devices because of the heavy computation. In recent years, a series of compact networks [4]–[9] are proposed with the increasing demand for a lightweight model. Based on a streamlined architecture, MobileNets [4] adopts depth-wise separable convolutions to establish lightweight deep neural networks and efficiently trades off between latency and accuracy. MobileNetV2 [5] has introduced a new mobile architecture that consists of inverted residuals and linear bottlenecks. Through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdpt algorithm, MobileNetV3 [6] takes advantage of the novel architecture to subsequently improve the accuracy. Besides, it further explores the issue of how automated searching algorithms and network design can work together to improve the overall state of the art on mobile classification, detection and segmentation. ShuffleNet [7] primarily introduces pointwise group convolution operations and channel shuffle operations to extensively decrease computational cost. Ma *et al.* [8] presented a new architecture called ShuffleNet V2 and their work derives several practical guidelines for efficient network design. Accordingly, comprehensive experiments have demonstrated that their work achieves state-of-the-art performance in terms of speed and accuracy tradeoff. Han *et al.* [9] proposed a novel Ghost module to generate more feature maps from cheap operations. They applied a series of linear transformations on these intrinsic feature maps to generate many relevant feature maps for getting primary information with a less computational cost.

D. ATTENTION MECHANISM

Recently, related works based on attention mechanism have achieved great success in various computer vision tasks such as image classification [9], [37], object recognition [38], lightweight human pose estimation [10], and so on. Chu *et al.* [36] firstly introduced attention mechanism into pose estimation models. They proposed a method that incorporates convolutional neural networks with a multi-context attention mechanism into an end-to-end framework. Non-local network proposed by Wang *et al.* [3] employs self-attention mechanisms to model pixel-level pairwise relations for capturing long-range dependencies. Although gaining some performance in human pose estimation, the non-local operation computes the response at each query position with extensive computation cost. Hu *et al.* [2] proposed a novel architectural unit, termed as Squeeze-and-Excitation (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. These blocks can be stacked together to form SENet that significantly improves the performance for top-performing models at a slight additional computational cost. Based on rigorous empirical analysis, Cao *et al.* [1] found that the global contexts modeled by the non-local network are almost the same for different query positions within an image. They designed a better instantiation, called the global context block (GCB) and constructed a global context network (GCNet), which maintains the performance of

NLNet but takes significantly less computational cost. Therefore, it is applied in each bottleneck block of our model that can increase the performance of our network without too much computational cost.

E. STRUCTURAL SIMILARITY

Generally, feature maps are highly structured in that their pixels exhibit strong relationships, especially when they are spatially and temporally proximate. Their relationships in spatial and temporal sequence usually carry extremely significant information about the structure of the object in visual scenarios. Wang *et al.* [39] constructed a formulation SSIM for measuring the structural similarity quality from the perspective of image formation. Their method composes of three parts: the average luminance, contract, and structural information.

Unlike [39], we just take two channels of the feature maps as the input of the method. Then, SSIM is adopted to evaluate the structural similarity among these feature maps which come from one original image. Finally, a value computed by SSIM indicates the similarity between two input feature maps and the larger value which ranges from 0 to 1 means a strong relationship. These output values determine the compress ratio of intrinsic feature maps in our module.

Despite their top performance, we will analyze the performance in Section V by comparing the size of parameters, GFLOPs, and inference time among these state-of-the-art methods. The aforementioned lightweight blocks aim to reduce the computational cost without too much accuracy decrease. However, their methods (LPN) have not been extensively used for human pose estimation and their performance has not been demonstrated by extensive experiments. Therefore, we propose a simple but powerful human pose estimation model with a lightweight bottleneck block which can significantly reduce the computational cost.

III. APPROACH

Owing to complex architecture and vast computational cost, a lightweight but powerful pose estimator is extremely hard to design which is described in Section I. To conquer this limitation, we propose a novel lightweight human pose estimation method by redesigning a simple network (FLPN) with several groups of lightweight bottleneck (Smart bottleneck) blocks. The smart bottleneck is mainly composed of two stacked smart modules and a global context (GC) [1] block. Then, the smart module is introduced to utilize cheap operations to generate more feature maps from these intrinsic feature maps. The structural similarity (SSIM) [39] measurement method is adopted in the smart module and determines the appropriate proportion of intrinsic feature maps in the total feature maps. At the same time, we also append the GC block which is effectively able to model the global context by capturing long-range dependencies with the less computational cost increase. To achieve extremely efficient architecture and high performance, we proposed the FLPN network with a simple architecture.

Firstly, we explain the architecture of the novel lightweight module, bottleneck and efficient network and then compare the computational cost of the proposed method.

A. SMART MODULE FOR MORE FEATURES

The success of GhostNet [9] proposed by Han et. al has provided prior knowledge that intermediate feature maps calculated by mainstream CNNs often contain mass redundancy and some of them are similar in many aspects. Inspired by their creative idea, we point out that it is necessary to compare feature maps between different channels for one input image and determine the ratio of intrinsic feature maps in the module.

Top-performing human pose estimation models [10], [13], [34] often adopt ResNet as their backbone with a large number of convolution layers that result in extremely massive computational cost. Given the comprehensively existing redundancy in the process of feature maps calculated by these high performance models as shown in Figure 1, Han *et al.* [9] proposed the ghost module to reduce the demanded resources. The ghost module adopts a handful of intrinsic feature maps to generate more feature maps with some cheap transformations. However, there is a question that why these intrinsic feature maps occupy half of the input maps. In this section, we will further explore the reason for the compression ratio.

In practice, $X \in R^{(c\times h\times w)}$ is the input data, where c is the number of input channels. At the same time, h and w represent the height and width of the input data, respectively. $Y \in R^{(n\times w'\times h')}$ is the output map with n channels. The operation of the convolutional layer, which transforms the input data X into output map Y, can be formulated as:

$$Y = X * f + b \tag{1}$$

where $*$ is the transformation operation, b is the bias term and $f \in R^{(c\times k\times k\times n)}$ is the convolution filters in convolutional layer. Besides, h' and w' are the weight and width of the output feature maps, and $k \times k$ is the kernel size of the convolution filters f, respectively.

During the convolutional process, a standard convolutional layer can be parameterized by convolution kernel k of the size $D_k \times D_k \times c \times n$, where D_k is the spatial dimension of the kernel assumed to be square. Correspondingly, the calculated number of FLOPs can be formulized as $n \times h' \times w' \times c \times k \times k$, which often results in massive computation cost because of the large number of filters n and abundant channel numbers c.

Based on Eq. 1, the large number of parameters in f and b to be optimized can be simplified to reduce the dimensions of input feature maps and output feature maps. We point out that the ratio of intrinsic feature maps dynamically change with the number of redundant feature maps in total feature maps. Therefore, we introduce SSIM to estimate the similarity among these feature maps and find the appropriate ratio of intrinsic feature maps. Specifically, m^d intrinsic feature maps $Y \in R^{(h'\times w'\times m^d)}$ are produced by some primary convolution filters:

$$Y^d = X \times f^d \tag{2}$$

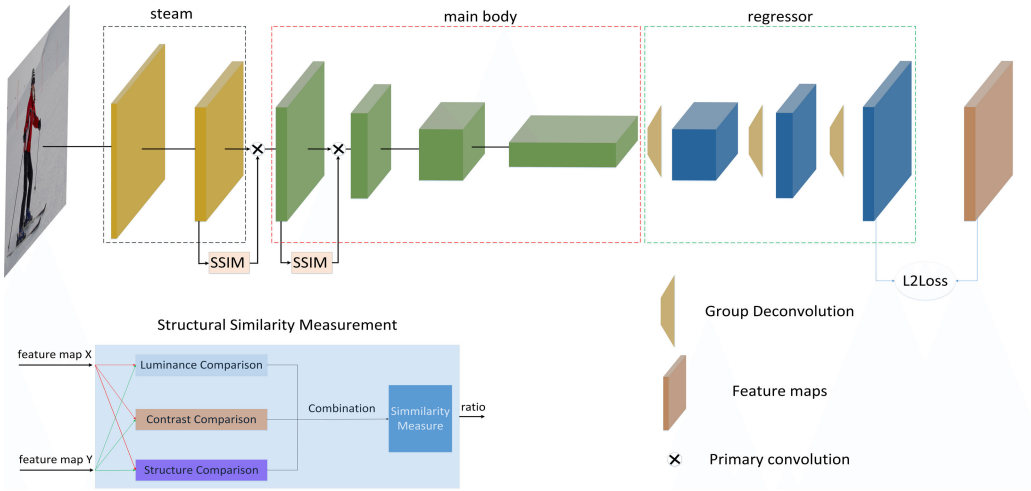


FIGURE 2. Illustrating the architecture of the FLPN network. Similar to SimpleBaseline [13], our method contains a stem, a backbone network and several up-sampling layers. Differently, we introduce the method of SSIM to evaluate the similarity among feature maps, redesign the bottleneck block (green blocks in the above figure.), and choose a novel lightweight fashion for up-sampling.

where $f \in R^{(c\times k\times k\times m^d)}$ is the convolution filters, m^d is much lower than n and the bias b is omitted for simplicity. To keep the consistent spatial size of the output feature maps, a series of cheap linear operations are adopted on each intrinsic feature maps in Y^d to generate t relevant feature maps in the following equation:

$$y_{i,j} = \phi_{i,j}(y'_i) \tag{3}$$

where y'_i is the i-th intrinsic feature maps in Y^d, $\phi_{i,j}$ is the j-th linear operation for generating the j-th ghost feature map $y_{i,j}$. Obviously, each y' can have t ghost feature maps. The final $\phi_{i,s}$ is an identity mapping for maintaining the intrinsic feature maps. Finally, we integrate these intrinsic feature maps and relevant feature maps into the output feature maps with a consistent spatial size. In terms of computational cost, these linear operations ϕ are much less than the standard convolution.

As shown in Figure 3, the blue block represents the method of SSIM which determines the ratio of intrinsical feature maps. Under the guidance of SSIM, we increase the number of linear operations which can reduce the computational cost in our proposed method. The identity represents the intrinsical feature maps and others generated by cheap operations (ϕ) represent relevant features. The above standard convolution is used to compare with ours.

As the module can be easily integrated into top-performing human pose estimation networks to reduce the computational cost, we further analyze the income on theoretical speed-up ratio and the total number of parameters. There exits one identity mapping and $m^d \times (t-1) = \frac{n}{t} \times (t-1)$ linear operations, and the averaged kernel size of each linear operation is equal to $d \times d$. For simplification, we take the same kernel size for linear operation and ordinary convolution layer in one module for efficient performance. The total number of parameters for an ordinary convolutional layer is $n \cdot k \cdot k \cdot c$. Comparatively, the

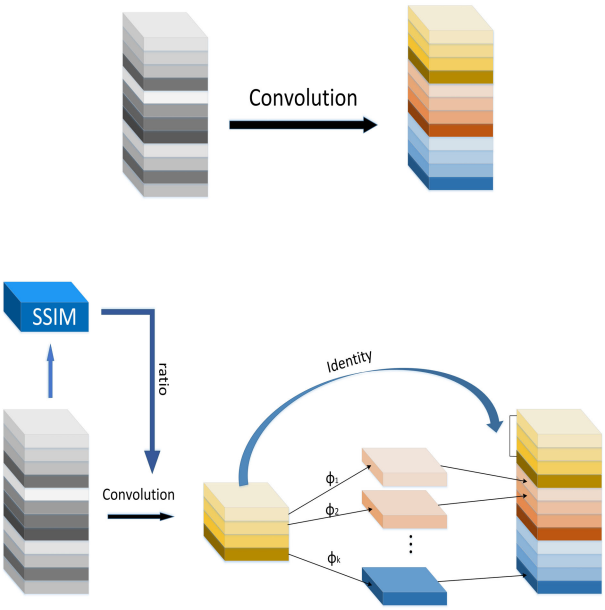


FIGURE 3. An illustration of the process of the proposed smart module for generating the same number of maps. The method of SSIM can determine the ratio of intrinsic feature maps. ϕ represents the cheap operation. Output feature maps consist of intrinsic feature maps and relevant feature maps calculated by a cheap operation.

parameters of our module compose of primary convolution $m \cdot k \cdot k \cdot c$ and linear operations $m \cdot k \cdot k \cdot (t-1)$. The compression ratio of parameters can be calculated as:

$$\begin{aligned} r_c &= \frac{n \cdot c \cdot k \cdot k}{m \cdot k \cdot k \cdot c + m \cdot k \cdot k \cdot (t-1)} \\ &= \frac{n \cdot c \cdot k \cdot k}{\frac{n}{t} \cdot k \cdot k \cdot c + \frac{n}{t} \cdot k \cdot k \cdot (t-1)} \\ &\approx \frac{t \cdot c}{t + c - 1} \approx t \end{aligned} \tag{4}$$

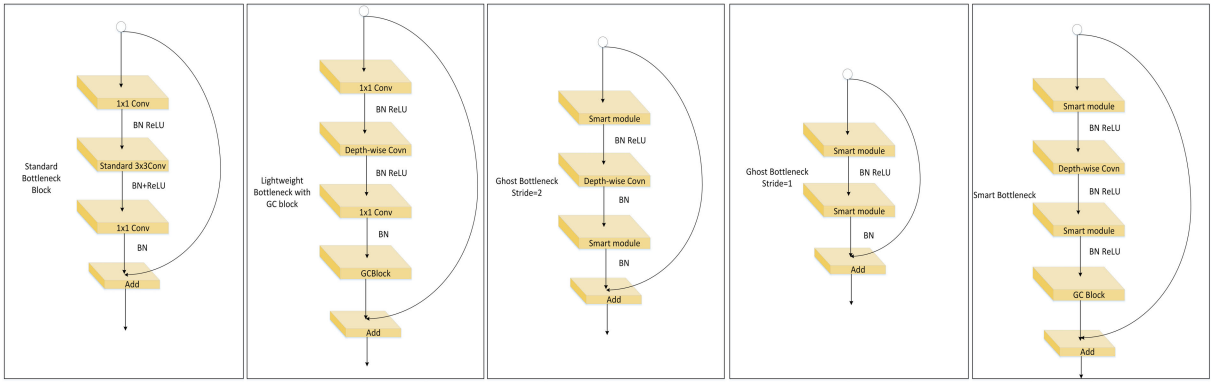


FIGURE 4. Left to right:(1) Standard bottleneck; (2) Lightweight Bottleneck; (3)(4) Ghost Bottleneck; (5) Smart Bottleneck.

Similarly, the theoretical speed-up ratio can be formulized as

$$\begin{aligned} r_c &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{m \cdot h' \cdot w' \cdot k \cdot k \cdot c + m \cdot k \cdot k \cdot (t-1) \cdot h' \cdot w'} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{t} \cdot k \cdot k \cdot c} + \frac{1}{t} \cdot k \cdot k \cdot (t-1) \\ &\approx \frac{t \cdot c}{t + c - 1} \approx t \end{aligned} \tag{5}$$

where $s \leq c$. In our paper, $t \geq 2$ leads to much decrease in computational cost.

B. BUILDING LIGHTWEIGHT BOTTLENECK BLOCK

The bottleneck is first introduced in ResNet [23]. As shown in Figure 4, the bottleneck block composes of several convolutional layers and a shortcut connection. Correspondingly, the total number of the parameters for a standard bottleneck block can be represented as

$$1 \times 1 \times N \times M + 3 \times 3 \times M \times M + 1 \times 1 \times M \times N \tag{6}$$

For a bottleneck, the number of input channels N is consistent with that of output channels and $N = M \times expansion$. Correspondingly, M represents the hidden dimensions and *expansion* is a hyperparameter with a default value of 4. Therefore, the above Eq.11 can be simplified as

$$17 \times M \times M \tag{7}$$

Based on three modifications of the standard bottleneck block, we introduce the novel bottleneck (Smart bottleneck) specially designed for lightweight networks. Taking the advantage of the lightweight module, we firstly replace the standard bottleneck with our smart module, which can significantly reduce the computation cost. And then, the standard 3×3 convolution is replaced by a 3×3 depth-wise convolution, which can generate more features with fewer parameters. Finally, it is illustrated in Figure 5, we also adopt the global context (GC) block in the lightweight bottleneck, which can capture long-range dependencies without too much computational cost. As shown in Figure 4, the structure of the bottleneck seems to be similar to the bottleneck

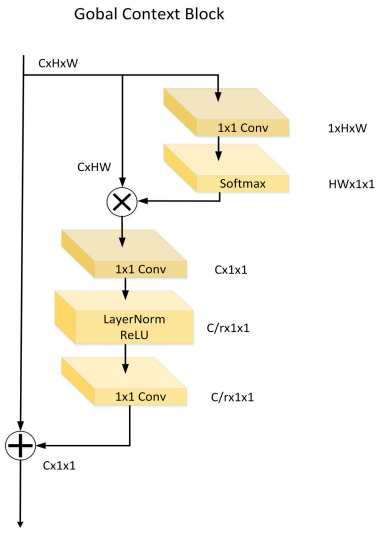


FIGURE 5. Global context block.

in ResNet. The most obvious difference with Ghost Bottleneck [9] is the application of the two stacked modules. Except for downsampling between stages, our bottleneck maintains the same number of channels in the stage. To a certain extent, for designing a lightweight bottleneck, we aim to reduce these operations between channels rather than increasing the number of channels. The number of parameters of the smart bottleneck is

$$\frac{2}{t} \times 1 \times 1 \times N \times M + 3 \times 3 \times M \approx \frac{8}{t} \times M \times M \tag{8}$$

where t is the compression ratio for a module. Thus, the final reduction in the parameter is

$$\frac{\frac{2}{t} \times 1 \times 1 \times N \times M + 3 \times 3 \times M}{17 \times M \times M} \approx \frac{8}{17t} \tag{9}$$

where t determined by SSIM in the range [2], [16].

C. FAST AND LIGHTWEIGHT NETWORK

The simple and widely adopted pipeline [10], [13] to estimate human pose consists of a stem decreasing the size of

input images, the main body learning the features of these maps by reducing the resolution continuously, and a regressor estimating the heatmaps by transforming these low resolution heatmaps into the full resolution heatmaps and choosing the accurate positions of key points. Following the simple design principle, SimpleBaseline [13], which achieves top-performing in human pose estimation, adopts a series of standard bottlenecks as the main body and employs several deconvolutional layers as the regressor. Inspired by their simple design architecture, we basically follow the architecture of SimpleBaseline [13] for its superiority and replace these standard bottleneck blocks used in the backbone with our smart bottleneck blocks. In the stem, we use two successive convolutional layers with a small kernel size (3×3) to reduce the resolution of input images rather than a convolution layer with a large kernel size (7×7) followed by a max-pooling layer. The main body consists of a series of bottlenecks with gradually increasing channel numbers and decreasing feature map resolution. These bottleneck blocks are grouped into four stages according to the input size of their feature maps. Under the guidance of the SSIM method, the appropriate ratio of these intrinsic feature maps in the module varies with the depth of the network. During the downsampling process, we experimentally demonstrate that the correlation among feature maps fades away gradually. It means that there exist a large number of redundant features in large size of feature maps and using a low ratio can drastically reduce the scale of parameters and FLOPs. Through the whole process, we replace these convolution layers with group convolution layers as many as possible to reduce the abundant parameters while keeping the quality of feature maps. Finally, the group size of the group convolutions can be simplified to the great common divisor of input channels and output channels. The architecture of our network is illustrated in Figure 2.

IV. IMPLEMENTATION

In this section, we mainly describe the training setup, two datasets which are publicly available benchmarks used for human pose estimation, and an evaluation protocol. Moreover, we also introduce evaluation metrics for every dataset.

A. TRAINING SETUP

The same set of parameters and settings as SimpleBaseline [13] and LPN [10] were adopted to guarantee a fair comparison between the two methods [10], [13] and our method. Our network and the above mentioned two networks were all initialized by pre-training on the ImageNet classification task [21]. The Adam Optimizer [22] was also adopted. Similar to the two methods, the base learning rate was initiated at 1e-3 and dropped to 1e-4 at 90 epochs and 1e-5 at 120 epochs respectively. These networks were trained for 140 epochs in total. Except for a similar network as SimpleBaseline, we also use the novel lightweight module and SSIM to fine tune the proposed network.

Following [10], [12], [13], the input image is cropped into a fixed ratio bounding box with the human. Then, we resize the bounding box to 256×192 to train our model. Moreover, data augmentation, composed of scale, rotation and filp, was applied to train the baseline methods [10], [13] and our proposed method. For the COCO2017 dataset [11], random rotation through $[-40, 40]$ degrees, random scalings in $[0.7, 1.3]$, and horizontal flips were adopted. For the MPII dataset [14], random rotation through $[-30, 30]$ degrees, random scalings in $[0.75, 1.25]$, and horizontal flips were also adopted. In the testing phase, we use human body detection bounding boxes based on COCO2017 to crop these images and put them into our model to evaluate the performance of our method. During the actual inference stage, a human body detector finds the human body box and puts it into our model to generate human poses.

B. DATASETS

1) COCO KEYPOINT DETECTION DATASET

The COCO dataset [11], widely used for human pose keypoint detection, contains over 200, 000 images and 250, 000 person instances labeled with 17 keypoints. Three datasets train2017/val2017/test-dev2017, cover 57K, 5K and 20K images individually, are used for training our model, evaluating our approach locally and evaluating our approach on an online platform respectively. Most of the existing methods [12], [13] evaluate the performance on 256×192 input images by cropping the heights and widths in a 4: 3 ratio, therefore, we trained our network to utilize 256×192 input images to ensure a fair comparison.

The mean average precision(AP) and average recall(AR) were adopted as evaluation metrics based on object keypoint similarity(OKS) to evaluate the result. The standard evaluation metric is based on Object Keypoint Similarity (OKS):

$$OKS = \frac{\sum_i exp(-d_i^2/2S^2k_i^2)\sigma(v_i > 0)}{\sum_i \sigma(v_i > 0)}. \tag{10}$$

OKS is a measure that converts the Euclidean distance d_i between the ground truth keypoint and the estimated keypoint to a value between 0 and 1. Here v_i indicates the visibility of the ground truth, s indicates the object scale, and k_i indicates a per-keypoint constant that controls falloff. AP^{50}(the average precision as OKS $= 0.50$), AP^{75}(the average precision as OKS $= 0.75$), AP(the mean of AP scores at OKS $=$ 0.5, 0.55, 0.6, …, 0.95), AR(the mean of average recall scores at OKS $= 0.5, 0.55, 0.6, \ldots, 0.95$). Further, AP^M for medium objects (object area between 32^2 and 96^2) and AP^L for large objects (object area larger than 96^2) were reported.

2) MPII HUMANPOSE ESTIMATION DATASET

The MPII Human Pose Dataset [14] composes of real-world images taken from various human daily activities with full-body pose annotations. This dataset contains over 25K images and 40K subjects, where 12K subjects are used for testing and the remaining subjects are used for training. The data augmentation and the training setup are the same as COCO2017, except that the size of the input image was

TABLE 1. Quantitative comparisons on the COCO2017 validation dataset.

Method	Backbone	Pretrain	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass	Hourglass	N	256×192	25.6M	26.2G	66.9	—	—	—	—	—
HigherHRNet	HRNet-W32	—	512×512	28.6M	47.9G	67.1	86.2	73.0	61.5	76.1	—
HigherHRNet	HRNet-W32	—	640×640	28.6M	74.8G	68.5	87.1	74.7	64.3	75.3	—
CPN	ResNet-50	Y	256×192	27.0M	6.2G	68.6	—	—	—	—	—
HigherHRNet	HRNet-W48	—	640×640	63.8M	154.3G	69.9	87.2	76.1	65.4	76.4	—
SimpleBaseline	ResNet-50	Y	256×192	34.0M	8.9G	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline	ResNet-101	Y	256×192	53.0M	12.4G	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline	ResNet-152	Y	256×192	68.6M	15.7G	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32	HRNet-W32	N	256×192	28.5M	7.1G	73.4	89.5	80.7	70.2	80.1	78.9
HRNet-W32	HRNet-W32	Y	256×192	28.5M	7.1G	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48	HRNet-W48	Y	256×192	63.6M	14.6G	75.1	90.6	82.2	71.5	81.8	80.4
LPN*	ResNet-50	N	256×192	2.9M	1.0G	69.1	88.1	76.6	65.9	75.7	74.9
LPN*	ResNet-101	N	256×192	5.3M	1.4G	70.4	88.6	78.1	67.2	77.2	76.2
LPN*	ResNet-152	N	256×192	7.4M	1.8G	71.0	89.2	78.6	67.8	77.7	76.8
FLPN(Ours)	SResNet-50	Y	256×192	10.04M	1.1G	71.3	91.6	79.0	68.8	75.3	74.5
FLPN(Ours)	SResNet-101	Y	256×192	17.0M	2.0G	72.6	92.5	80.4	70.1	76.6	75.8
FLPN(Ours)	SResNet-152	Y	256×192	22.5M	2.8G	73.1	92.6	80.4	70.8	76.7	76.3

The entries with the performance and the computational cost are clearly illustrated in Table 1. The method with "*" denotes that it adopts an iterative strategy on its own model rather than pretrains its backbone on the ImageNet classification task. Our model saves a great lot of computational cost and maintains top-performing overall compared approaches.

cropped to 256×256 for providing a fair comparison with other methods.

The PCKh (head-normalized probability of correct keypoint) score is adopted as the standard evaluation metric in MPII human pose estimation. A joint is correct if it falls within αl pixels of the ground-truth position, where α is a constant and l is the head size that corresponds to 60% of the diagonal length of the ground-truth head bounding box. PCKh@0.5 is used for evaluating the accuracy of joint localization, which indicates that the distance between the estimation joint point and the ground-truth is less than 0.5 times the length of the head segment.

C. EVALUATION PROTOCOL

The general accuracy evaluation metrics were applied in the proposed method for a fair comparison with other methods. Apart from that, we redesigned experiments to measure the performance of our proposed lightweight method with state-of-the-art human pose estimation methods. These experiments were divided into the following three parts for detailed analysis.

- Experiment 1: To be more general and fair, we compare our method with state-of-the-art methods on the two publicly available benchmarks: COCO2017 and MPII. Besides, another major task was to explore efficient network which occupying low resource and achieving high accuracy. The selected simple method will take part in the next experiment.
- Experiment 2: Lots of lightweight models applied for image classification make the human pose estimation network available for mobile devices. In terms of inference time, not all top-performance methods suit mobile edge devices. Considering low calculation cost and high performance, we use a lightweight bottleneck block to replace the bottleneck of the selected model and compare their performance with ours.
- Experiment 3: Intuitionally, the size of the model is an extremely significant factor for evaluating the performance of the model. Therefore, we adopted SSIM to fine tune the proposed network and find the appropriate ratio which could balance the accuracy and lightweight size.

Under the guidance of the evaluation protocol, we can fairly compare our proposed method with others on the performance, calculational cost, and the size of parameters. In the next section, we will use quantitative results and qualitative results to demonstrate the performance of our method.

V. EXPERIMENTS

A. EVALUATING PERFORMANCE

We compare our method with various top-performing methods on the COCO2017 dataset and the MPII dataset. For the fair comparison, we adopt the same person detector provided by HRNet [12] that can evaluate the inference time of these methods based on a uniform criterion. It is reasonable to compare our method with these top-performing methods under the above evaluation protocols in Section IV. For these lightweight models whose official codes are not available, we directly adopt their published data for comparisons.

1) COMPARISONS WITH SOTA METHODS

The results which present the performance comparisons under the above mentioned protocol1 are summarized in Table 1 and Table 2. Notably, we point out that iterative training on his own pre-training model can increase their accuracy with a lot of time consuming. The lightweight pose method LPN has not released their codes online. Therefore, we adopt the reproducible version of LPN for comparison

TABLE 2. Quantitative comparisons on the COCO2017 test-dev dataset.

Method	Backbone	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Bottom-up: keypoint detection and grouping										
Openpose	—	—	—	—	61.8	84.9	67.5	57.1	68.2	66.5
Associative	—	—	—	—	65.5	86.8	72.3	60.6	72.6	70.2
HigherHRNet	HRNet-W32	512×512	28.6M	47.9G	66.4	87.5	72.8	61.2	74.2	—
HigherHRNet	ResNet-50	640×640	63.8M	154.3G	68.4	88.2	75.1	64.4	74.2	—
PersonLab	—	—	—	—	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet	—	—	—	—	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: Human detection and single-person keypoint detection										
Mask-RCNN	ResNet-50-FPN	—	—	—	63.1	87.3	68.7	57.8	71.4	—
G-RMI	ResNet-101	353×257	42.6M	57.0G	64.9	85.5	71.3	62.3	70.0	69.7
Integral Regression	ResNet-101	256×256	45.0M	11.0G	67.8	88.2	74.8	63.9	74.0	—
G-RMI+extra data	ResNet-101	353×257	42.6M	57.0G	68.5	87.1	75.5	65.8	73.3	73.3
SimpleBaseline	ResNet-50	256×192	34.0M	8.9G	70.0	90.9	77.9	66.8	75.8	75.6
SimpleBaseline	ResNet-101	256×192	68.6M	15.7G	71.6	91.2	80.1	68.7	77.2	77.3
CPN	ResNet-Inception	384×288	—	—	72.1	91.4	80.0	68.7	77.2	78.5
RMPE	PyraNet	320×256	28.1M	26.7G	72.3	89.2	79.1	68.0	78.6	—
CFN	—	—	—	—	72.6	86.1	69.7	78.3	64.1	—
CPN(ensemble)	ResNet-Inception	384×288	—	—	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline	ResNet-152	384×288	68.6M	35.6G	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32	HRNet-W32	384×288	28.5M	16.0G	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48	HRNet-W48	384×288	63.6M	32.9G	75.5	92.5	83.3	71.9	81.5	80.5
HRNet-W48	HRNet-W48	384×288	63.6M	32.9G	77.0	92.7	84.5	73.4	83.1	82.0
LPN*	ResNet-50	256×192	2.9M	1.0G	68.7	90.2	76.9	65.9	74.3	74.5
LPN*	ResNet-101	256×192	5.3M	1.4G	70.0	90.8	78.4	67.2	75.4	75.7
LPN*	ResNet-152	256×192	7.4M	1.8G	70.4	91.0	78.9	67.7	76.0	76.2
FLPN(Ours)	SResNet-50	256×192	10.04	1.1G	68.7	90.6	77.2	65.9	74.0	74.5
FLPN(Ours)	SResNet-101	256×192	17.0M	2.0G	69.9	90.9	78.4	67.2	75.3	75.7
FLPN(Ours)	SResNet-152	256×192	22.5	2.8G	70.3	91.0	78.9	67.6	75.6	76.1

Human pose estimation methods are split into two types (Bottom-up and Top-down). Our method belongs to Top-down method which depends on the human body detector to capture the box around the human.

without iterative training and the results are much lower than these top-performing methods.

From the results of Table 1, our methods have been achieved comparable performance with the SimpleBaseline series and HRNet series. On the COCO2017 validation dataset, our methods surpass SimpleBaseline on various backbones (e.g. Resnet50, Resnet101, and Resnet152). Especially, the size of parameters and flops of our proposed method whose backbone is Resnet50 are less than one-third and one-eighth of Simplebaseline (Resnet50) respectively. Though achieving lower accuracy compared with the HRNet series, the size of parameters and flops of our methods much lower than theirs. Obviously, the parameter size of HRNet-W32 is more than three times of our method (Resnet50). At the same time, the size of the flops is six times that of ours. On the COCO2017 test-dev set, our method achieves comparable results with these top-performing methods with the same input image size of 256×192. Even LPN adopts an iterative training strategy, our lightweight method also achieves the same performance as the LPN series.

In summary, our proposed method significantly outperforms all state-of-the-art methods in the size of parameters and computational cost. At the same time, our model maintains a similar accuracy with these top-performing methods. In comparison with the LPN, we have a large size of parameters and similar computational cost, but the accuracy of our method is greater than their method.

To further compare our method with these top-performing methods, we trained our model on the MPII dataset and evaluated the performance. The results are described in Table 3 and Table 4. Different from the above mentioned experiments, the size of the input image is 256×256. On the MPII val dataset, our method achieves similar performance with much less computational cost. The small size of params and flops and high accuracy demonstrate the efficiency of our method.

2) COMPARISONS ON VARIOUS BOTTLENECK BLOCKS

Considering the computational cost and inference time of these top-performing methods presented in the above Table 1 and Table 3, we finally chose the SimpleBaseline (Resnet50) as the optimal model which can balance the accuracy and real-time performance. To further analyze the performance of various bottleneck blocks, we conduct some experiments on the COCO2017 dataset. For example, SimpleBaseline with ghost bottleneck block, SimpleBaseline with mobilenet-v3 bottleneck block are trained and compared with our method in Table 5 under the same experiments setting.

Ideally, the integration of SimpleBaseline and lightweight bottleneck is the best way for human pose estimation.

TABLE 3. Quantitative comparisons on the MPII val dataset (PCKh@0.5).

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Total
SimpleBaseline-50	96.4	95.3	89.0	83.2	88.4	84.0	79.6	88.5
SimpleBaseline-101	96.9	95.9	89.5	84.4	88.4	84.5	80.7	89.1
SimpleBaseline-152	97.0	95.9	90.0	85.0	89.2	85.3	81.3	89.6
HRNet-W32	97.1	95.9	90.3	86.4	89.1	87.1	83.3	90.3
FLPN-50(Ours)	96.3	94.9	88.0	81.7	88.1	83.0	78.1	87.8
FLPN-101(Ours)	96.6	95.2	88.7	83.1	88.0	83.5	79.2	88.3
FLPN-152(Ours)	96.2	95.2	88.6	82.7	88.4	83.6	80.0	88.4

In the training phase, flip test is used in all above experiments. The input size of images is 256×256.

TABLE 4. Quantitative comparisons on the #Params and FLOPs of some excellent methods.

Method	Backbone	#Params	FLOPs	PCKh@0.5
SimpleBaseline	ResNet-50	34.0M	9.7G	88.5
SimpleBaseline	ResNet-101	53.0M	13.3G	89.1
SimpleBaseline	ResNet-152	68.6M	17.0G	89.6
HRNet-W32	HRNet-W32	28.5M	9.5G	92.3
FLPN	SResNet-50	10.0M	1.1G	87.8
FLPN	SResNet-101	16.9M	2.0G	88.3
FLPN	SResNet-152	22.5M	2.8G	88.4

The #Params and FLOPs are computed with the input size 256×256. Consider the actually inference time and FPS, we chose the lightweight human pose estimation network to compare their accuracy and their computational cost consists of the size of parameters and float-point operations.

However, we adopted ResNet50 as backbone and mobilenetv3-bottleneck in the same experimental setting. Finally, the SimpleBaseline with ghostnet-bottleneck performs better than SimpleBaseline with mobilenetv3-bottleneck and original SimpleBaseline. Notably, our method outperforms all these methods with an accuracy of 71.3. Our method has a similar size to SimpleBaseline with ghostnet bottleneck and the size of flops is rather less than these methods with various bottlenecks.

3) COMPARISONS ON INFERENCE SPEED

To compare the inference performance of our method and these compared methods, we have conducted all the experiments on the same platform that composes of an Intel 2.8GHz CPU and one NVIDIA GeForce GTX 1080Ti GPU. In this section, we mainly compare the inference time for these top-performing methods. The inference time consists of detecting human body boxes and estimating human keypoints. In our experiments, we adopt the same human detector to detect human body boxes and the inference time is about 2.5 seconds. The inference time of estimating human keypoints changes with different methods. As described in Figure 6, our method achieves the fastest speed among these methods. According to the appropriate reference [40]–[42], we also concect all the predicted key points in Figure 7. The results confirm the efficiency of the proposed method.

B. ABLATION STUDY

Ablation study is conducted to analyze the effect of each component in our methods, including the lightweight bottleneck block, the redesigned Network, and the method of SSIM. Under the above mentioned protocol in Section IV, our method was trained on variant conditions and we conducted extensive experiments on the COCO2017 dataset to pursue a detailed component analysis.

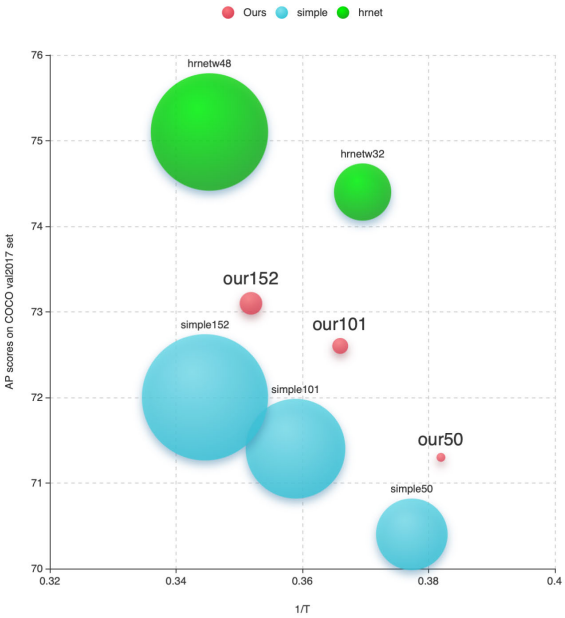


FIGURE 6. Comparisons of AP score, speed and FLOPs of ours and these top-performing methods referred in Table 1 on a non-GPU platform. Notably, we adopt the same input size 256 × 192 for all experiments. Several different colors denote the same backbone with various bottleneck blocks. The area of a circle represents the size of the FLOPs.

1) LIGHTWEIGHT BOTTLENECK BLOCK

To demonstrate the superiority of the lightweight bottleneck block, we build our model by utilizing the lightweight bottleneck block without and with GC block respectively and compare them with the original experiment on the same platform. The following experiments were conducted on the COCO2017 validation dataset: utilizing the Smart bottleneck block and the Smart bottleneck block without a GC block in both training and inference process for estimating the final heatmaps (denoted as "Ours" and "Ours w / o GC block" respectively).

The result in Table 8 demonstrates that our method outperforms ours w / o GC block (71.3 percent versus 69.1 percent). In terms of the size of parameters and float-point

TABLE 5. Quantitative comparisons on the COCO2017 validation dataset.

Method	Bottleneck block	Pretrain	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline	standard	Y	256×192	34.0M	8.9G	70.4	88.6	78.3	67.1	77.2	76.3
Simplebaseline	Mobilenetv3-bottleneck	Y	256×192	45.7M	12.6G	61.2	86.3	67.4	59.5	64.1	64.7
SimpleBaseline	Ghostnet bottleneck	Y	256×192	7.4M	20.6G	71.1	91.5	79.2	68.3	75.4	74.3
Ours	Smart bottleneck	Y	256×192	10.2M	1.26G	71.3	91.6	79.0	68.8	75.3	74.5

In the comparison of lightweight bottleneck block, we adopt SimpleBaseline with ResNet-50 as our network for its top performance and high inference speed. The input size of the image is fixed to 256×256. All these methods are pretrained on the ImageNet classification task.

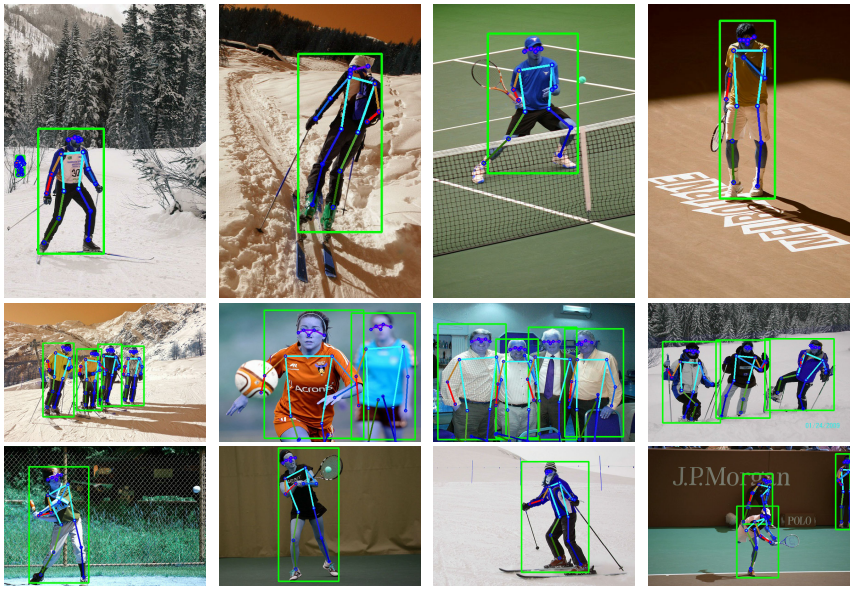


FIGURE 7. The prediction results of the proposed method on the COCO2017 dataset.

TABLE 6. Comparing the contribution of each element on COCO2017 validation dataset.

Method	Backbone	Pretrain	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Ours w / o beginning block	ResNet-50	Y	256×192	8.7980M	1.2521G	69.3	90.6	77.1	67.1	73.1	72.7
Ours w / o SSIM	ResNet-50	Y	256×192	8.9508M	1.3137G	69.3	90.5	77.0	67.0	73.0	72.5
Ours w / o group deconvolution	ResNet-50	Y	256×192	17.7887M	3.5253G	69.6	90.6	78.1	67.2	73.3	72.8
Ours	ResNet-50	Y	256×192	10.1922M	1.2638G	71.3	91.6	79.0	68.8	75.3	74.5

In the ablation study, we compare the contribution of each element to the accuracy of the network. All these experiments adopt the same size of input size and use pretrained model on the ImageNet dataset. "w /" represents adopt this method. "w / o" represents without this method.

operations (FLOPs), our method have a small increase than ours w / o GC block (about 2.6818M and 0.1277G). These results justify the contribution of the GC block without too much computational cost. Compared with the standard convolution bottleneck block (ours w / standard bottleneck), our method lower than ours w / standard bottleneck around 0.7 percent. However, the size of parameters and float-point operations are 2.3 times and 3.1 times that of our method. These results confirm that our method has a better ability to balance the accuracy and computational cost.

To further compare the performance of the Lightweight bottleneck block, we directly employ two kinds of state-of-the-art lightweight bottleneck blocks to replace our proposed bottleneck block. We denote the version of our model as "ours w / ghost bottleneck" and "ours w / mobilenet-v3" respectively. Table 8 demonstrates that our method performs better than ours w / ghost bottleneck (71.3 percent versus 69.7 percent) with similar or less computational cost (in terms of the size of parameters, 10.1922M versus 10.1594M), (in terms of the size of float-point operations, 1.2638G versus 1.7198G). As depicted in Table 6, our method significantly outperforms ours w / mobilenet-v3 bottleneck (69.6 percent versus 68.1 percent), and our module requires quite less computational cost. Considering the accuracy and computational cost, our method achieves the best balance between them as illustrated in Table 8.

2) PROPOSED NETWORK

The proposed network with different versions in the training and testing phase is illustrated in Table 6. Note that the max-pool layer may reduce some useful information for human pose estimation. Hence, we use a beginning block that contains two sequential convolutional layers to replace the original max-pool layer and denotes this version as

TABLE 7. Results for different intrinsical feature map ratio, depending on the method of SSIM.

stage	ratio				#Params	FLOPs	AP	AP^{50}	AR
	2	4	8	16					
stage0	√				8.59M	1.31G	69.3	90.5	72.5
stage1	√								
stage0		√			8.94M	1.29G	70.7	91.5	74.0
stage1	√								
stage0		√			8.84M	1.20G	70.8	91.5	74.0
stage1		√							
stage0			√		10.20M	1.26G	71.3	91.6	74.5
stage1	√								
stage0			√		8.83M	1.17G	69.0	90.5	72.3
stage1		√							
stage0			√		8.79M	1.15G	69.0	90.5	72.4
stage1			√						
stage0				√	10.19M	1.25G	70.8	91.6	74.1
stage1	√								
stage0				√	10.09M	1.18G	70.9	91.6	74.1
stage1		√							
stage0(Ours)				√	10.19M	1.26G	71.3	91.6	74.5
stage1(Ours)			√						
stage0				√	8.77M	1.12G	69.5	91.5	72.8
stage1				√					

TABLE 8. Results for COCO2017 validation dataset in a simple and lightweight network, depending on the type of bottleneck block.

Method	Backbone	Pretrain	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Ours w / ghost bottleneck	ResNet-50	Y	256 × 192	10.1594M	1.7198G	69.7	90.5	78.0	67.4	73.5	73.0
Ours w / mobilenet-v3 bottleneck	ResNet-50	Y	256 × 192	14.7798M	2.5543G	68.1	90.5	75.2	65.6	72.4	71.4
Ours w / standard bottleneck	ResNet-50	Y	256 × 192	23.5465M	3.9537G	72.0	91.6	79.3	69.5	76.1	75.1
Ours w / o GC block	ResNet-50	Y	256 × 192	7.5104M	1.1361G	69.1	90.5	77.2	66.9	72.8	72.5
Ours	ResNet-50	Y	256 × 192	10.1922M	1.2638G	71.3	91.6	79.0	68.8	75.3	74.5

"ours w / o beginning block". Compared with our model, the max-pool layer reduces the performance by around 0.3 percent. Then, to evaluate the effect of SSIM, we discard the SSIM method in our network which denoted as "Ours w / o SSIM". As illustrated in Table 6, our network with SSIM can have high performance than "Ours w / o SSIM". Most importantly, the size of float-point operations is reduced by SSIM. From the result of Table 6, we infer that expanding the ratio of cheap operation in the basic module can reduce the computational cost and increase performance. To further reduce the computational cost and maintain the high performance, group deconvolution is applied in the regression phase.

3) STRUCTURAL SIMILARITY MEASUREMENT

In our former experiments, we have found that the similarity of different channel feature maps comes from one image changes with the stage of our network. The structural similarity substantially decreases in the down-sampling stage0 and stage1. Meanwhile, the other two stages still maintain a lower level. Therefore, we adopt the first two stages and fixed ratios to explore the ideal model. Under the guidance of SSIM, we have employ ten group data to evaluate the performance of our model.

As illustrated in Table 7, to simplify the compression process and make the compression rate suit for our network, we use a group number (i.e 2, 4, 8, 16) to denote the proportion of the intrinsic feature maps. Thanks to the proposed structural similarity measurement mechanism, our model can effectively leverage the power of lightweight bottleneck block to decline the computational cost and maintain high performance. In Table 7, we compare ten versions of our model to determine the best one as our model.

VI. CONCLUSION

This paper presents a fast and lightweight method consists of FLPN network for more accurate pose estimation, a Smart bottleneck block for reducing the computational cost, and the method of SSIM to refine the appropriate ratio of intrinsic feature maps for reducing the module block size and maintaining the high accuracy. Extensive experiments on these above mentioned datasets demonstrate that our method has achieved similar accuracy with these top-performing methods and our computational cost is extremely lower than theirs. Considering the inference time and computational cost, our method is more suitable to employ edge devices. Finally, we hope our method could take some inspired ideas on real-time and lightweight pose estimation field.

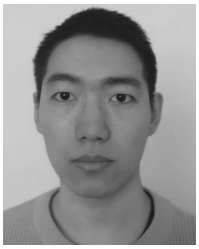
REFERENCES

[1] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCV)*, Oct. 2019, pp. 1971–1980.

[2] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.
[3] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.
[4] 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," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861
[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.
[6] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[7] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856.
[8] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[9] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1580–1589.
[10] Z. Zhang, J. Tang, and G. Wu, "Simple and lightweight human pose estimation," 2019, *arXiv:1911.10346*. [Online]. Available: http://arxiv.org/abs/1911.10346
[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision–ECCV 2014*. Cham, Switzerland: Springer, 2014, pp. 740–755.
[12] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.
[13] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.
[14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3686–3693.
[15] U. Iqbal, A. Milan, and J. Gall, "PoseTrack: Joint multi-person pose estimation and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2011–2020.
[16] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "DeepCut: Joint subset partition and labeling for multi person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4929–4937.
[17] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, pp. 1–12, May 2016.
[18] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "CrowdPose: Efficient crowded scenes pose estimation and a new benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10863–10872.
[19] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
[20] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1653–1660.
[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[24] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D pose estimation and action recognition using multitask deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5137–5146.
[25] G. Cheron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3218–3226.
[26] N.-G. Cho, A. L. Yuille, and S.-W. Lee, "Adaptive occlusion state estimation for human pose tracking under self-occlusions," *Pattern Recognit.*, vol. 46, no. 3, pp. 649–661, Mar. 2013.
[27] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Computer Vision—ECCV 2016*. London, U.K.: Elsevier, 2016, pp. 561–578.
[28] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1227–1236.
[29] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3D hand shape and pose estimation from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10833–10842.
[30] G. Moon, J. Y. Chang, and K. M. Lee, "PoseFix: Model-agnostic general human pose refinement network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7773–7781.
[31] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1799–1807.
[32] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision—ECCV 2016*. New York, NY, USA: Curran Associates, 2016, pp. 483–499.
[33] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2353–2362.
[34] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7103–7112.
[35] A. Bulat and G. Tzimiropoulos, "Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3706–3714.
[36] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1831–1840.
[37] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.
[38] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," 2014, *arXiv:1412.7755*. [Online]. Available: http://arxiv.org/abs/1412.7755
[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[40] S.-T. Kim and H. J. Lee, "Lightweight stacked hourglass network for human pose estimation," *Appl. Sci.*, vol. 10, no. 18, p. 6497, Sep. 2020.
[41] D. Freire-Obregón, M. Castrillón-Santana, P. Barra, C. Bisogni, and M. Nappi, "An attention recurrent model for human cooperation detection," *Comput. Vis. Image Understand.*, vols. 197–198, Aug. 2020, Art. no. 102991.
[42] P. Barra, C. Bisogni, M. Nappi, D. Freire-Obregón, and M. Castrillón-Santana, "Gender classification on 2D human skeleton," in *Proc. 3rd Int. Conf. Bio-Eng. Smart Technol. (BioSMART)*, Apr. 2019, pp. 1–4.

HAOPAN REN received the B.S. degree in mining engineering from Henan Polytechnic University, Henan, China, in 2018. Since 2018, he has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His current research interests include machine learning, computer vision, object detection, and human and hand pose estimation.

WENMING WANG received the master's degree from the School of Software, Hunan University, Hunan, China, in 2005. Since 2005, he has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His current research interests include machine learning, computer vision, object detection, and human and hand pose estimation.



KAIXIANG ZHANG received the B.S. degree from the School of Software Engineering, Xi'an University of Technology, China, in 2019. Since 2019, he has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His current research interests include machine learning, computer vision, object detection, human pose estimation and generation, and style transfer.

DEJIAN WEI received the B.S. degree from the School of Software, Shandong University, Shandong, China, in 2018. Since 2018, he has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His current research interests include machine learning, computer vision, object detection, and human and hand pose estimation.

YANYAN GAO received the B.S. degree from the School of Computer Science and Technology, OEC, Hebei, China, in 2009. Since 2018, he has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His current research interests include embedded systems, computer vision, and object detection.

YUE SUN received the B.S. degree from the School of Computer Science and Technology, Anhui University of Technology, Anhui, China, in 2019. Since 2019, she has been working with the Human-Computer Interaction Laboratory, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Her current research interests include computer vision, machine learning, and facial expression recognition.