Efficient High-Resolution Human Pose Estimation

Xiaofei Qin[1], Lingfeng Qiu[1], Changxiang He[2], and Xuedian Zhang[1(✉)]

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, 516 Jungong Road, Yangpu, Shanghai 200093, China
xiaofei.qin@usst.edu.cn, obmmdzxd@163.com

[2] College of Science, University of Shanghai for Science and Technology, 516 Jungong Road, Yangpu, Shanghai 200093, China

Abstract. As a fundamental task of computer vision, human pose estimation (HPE) has achieved significant improvement with the rise of deep learning. However, many existing methods focus too much on model accuracy, leading to high complexity models, which are hard to be deployed especially in computation-limited devices. This paper proposes a lightweight HPE network named efficient high-resolution human pose estimation (EHR-HPE). EHR-HPE network first adopts the high-resolution pattern to acquire accurate heatmaps; then an efficient shuffle block is proposed to reduce the model complexity and boost model performance; finally, the efficient dense connections are designed to further improve model accuracy. Extensive experiment results on two benchmark datasets show that the proposed EHR-HPE network achieves a great tradeoff between accuracy and model complexity. EHR-HPE network can achieve 70.1 mean average precision scores on Common Objects in Context (COCO) test-dev dataset with only 1.7M parameters and 0.91 GFLOPs.

Keywords: Human pose estimation · Lightweight network · Efficient shuffle block · Dense connection

1 Introduction

Human pose estimation (HPE) task is to detect and localize body keypoints (elbows, wrists, knees, etc.) of the input person images. It is a fundamental yet challenging task in the field of computer vision and is widely adopted for action recognition, pose tracking, human-computer interaction, etc.

HPE can be divided into single-person pose estimation and multi-person pose estimation according to the number of human instances in the input image. This paper focuses on single-person pose estimation because it is the basis for related vision tasks, such as multi-person pose estimation, video-based pose estimation, and pose tracking. Significant progress has been made in the field of HPE due

S. Khanna et al. (Eds.): PRICAI 2022, LNCS 13631, pp. 383–396, 2022.
https://doi.org/10.1007/978-3-031-20868-3_28

to the widespread use of deep neural networks [3,7,16,21,22,24]. These state-of-the-art methods typically employ deep and wide networks with a large number of parameters and a huge amount of floating-point operations (FLOPs), which result in high memory requirements and serious latency. These complex models are hard to be deployed especially for computation-limited devices (such as smartphones and embedded devices). Therefore, it is necessary to develop lightweight yet capable HPE methods.

Instead of recovering the resolution of input representation through a low-to-high process, HRNet [21] has a branch that maintains the highest resolution of the input representation throughout the process, this pattern is called high-resolution pattern in HRNet. The high-resolution pattern and multi-level features fusion strategies make HRNet becomes an excellent backbone for several vision tasks. However, the complexity of HRNet is very high. Small HRNet[1] is a much lighter network by reducing the depth and width of the original HRNet, but its performance drops significantly. This paper designed an efficient shuffle block to replace the costly residual block in Small HRNet, further reducing model complexity but improving performance. Furthermore, we added some efficient dense connections between the adjacent modules in the same stage to encourage feature reuse, which can also improve the performance of the model. This paper follows the high-resolution pattern in HRNet and Small HRNet, and considering the use of efficient shuffle block and efficient dense connections in it, this paper is consequently named efficient high-resolution human pose estimation (EHR-HPE) which architecture is shown in Fig. 1.

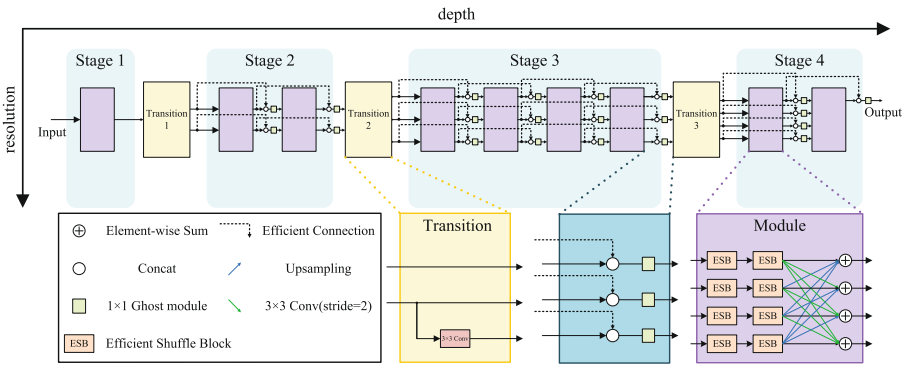

Fig. 1. The architecture of efficient high-resolution human pose estimation (EHR-HPE) network. The network consists of four stages, the deeper stage has more branches. Stages are connected by the transition layer that adds a new branch with lower resolution but more channel numbers through a 3×3 convolution. A stage has a sequence of modules, each module contains two efficient shuffle blocks for one branch and a multi-resolution fusing operation across the branches. The dashed lines indicate efficient dense connections between adjacent modules within a stage.

[1] Small HRNet: https://github.com/HRNet/HRNet-Semantic-Segmentation.

Experiments are conducted on two benchmark datasets to demonstrate the effectiveness and efficiency of the proposed EHR-HPE network. Experimental results show that the EHR-HPE network can achieve superior performance while maintaining a small model size and computation cost. The contributions of this paper can be summarized as follows:

1. The EHR-HPE network follows the high-resolution pattern of Small HRNet, and replaces the costly residual blocks with the newly designed efficient shuffle blocks. Efficient shuffle blocks adopt 1×1 Ghost modules [6] instead of costly pointwise convolutions to exchange information across channels. Additionally, an attention module is designed based on the GC block [2] to enhance the ability to model long-range dependencies and inter-channel relationships. Ablation studies have shown that the efficient shuffle block can achieve superior performance while halving computation cost and slightly reducing the number of parameters.
2. Efficient dense connections are added between the adjacent modules within the same stage, which inevitably widens the feature channels, so 1×1 Ghost modules are used to reduce the channel dimension. Efficient dense connections can strengthen feature reuse, facilitate convergence and promote network accuracy.

2 Related Works

In HPE tasks, the traditional methods adopt pictorial structure models [5,18,27] to infer the human pose. While these methods can perform efficient inference on simple images, they cannot handle complex scenarios such as occlusion. With the rise of deep learning, CNN-based methods [3,7,16,17,21,23,24,26] have become the main solution for HPE.

CNN-based HPE methods can fall into two categories, i.e., regressing the position of keypoints directly and estimating the keypoints through heatmaps. Compared with the direct regression method, the method based on heatmap [16,21,24] can fully use of the spatial information in the image, and achieve better accuracy and robustness. CPM [24] utilizes a multi-stage network to gradually refine detection results and adopts intermediate supervision to alleviate the vanishing-gradient problem. Hourglass network [16] follows the multi-stage pattern of CPM, and designs a symmetric encoder and decoder structure with short connections between the downsampling and upsampling branches to integrate multi-scale features. HRNet [21] further exploits the benefits of multi-scale features fusion by connecting sub-networks of different resolutions in parallel, preserving high-resolution features while fusing multi-level semantics to gain more accurate and precise heatmap estimation. The results of extensive experiments demonstrate the excellent performance of HRNet for HPE tasks. The method proposed in this paper follows the high-resolution pattern of HRNet, but contrives to give a more lightweight version.

Lightweight networks have aroused pervasive enthusiasm in the HPE research community. Zhang *et al.* [30] proposed LPN, which applies depthwise convolution

and attention mechanism to SimpleBaseline [26]. Qin *et al.* designed a lightweight HPE network named CVC-Net [19] based on pruned Hourglass, and many off-the-shelf tricks are used to enhance model performance. Based on HRNet, Yu *et al.* [28] proposed the conditional channel weighting unit to replace the costly pointwise convolution, which can greatly improve the computational efficiency at a slight decrease in accuracy.

Attention mechanism [11,12,25] can be regarded as a kind of conditional weight generation. Cao *et al.* [2] designed GC block to capture long-range dependencies. This paper designed a channel global context block named CGC, and combined CGC with GC block to form a more powerful attention module.

To capture more information across multiple scales, deep learning networks are now designed to go deeper, but vanishing-gradient problem occurs as the network deepens. ResNets [8] and DenseNet [13] build short paths between layers to alleviate vanishing-gradient, in which DenseNet realizes features reuse through concatenating feature maps with all subsequent layers and achieves better performance on several public datasets with fewer parameters. To strengthen the feature reuse of the HPE network, this paper follows the dense connection pattern of DenseNet to establish short paths between the adjacent modules within the same stage of the high-resolution architecture.

3 Method

3.1 Efficient Shuffle Block

Replacing Costly 1×1 Convolution. Efficient shuffle block is designed based on the shuffle block in ShuffleNet [15]. The shuffle block uses 1×1 convolutions to exchange information across channels, which is very costly and dominates the parameter and computational complexity of the shuffle block. As Fig. 2 (b)

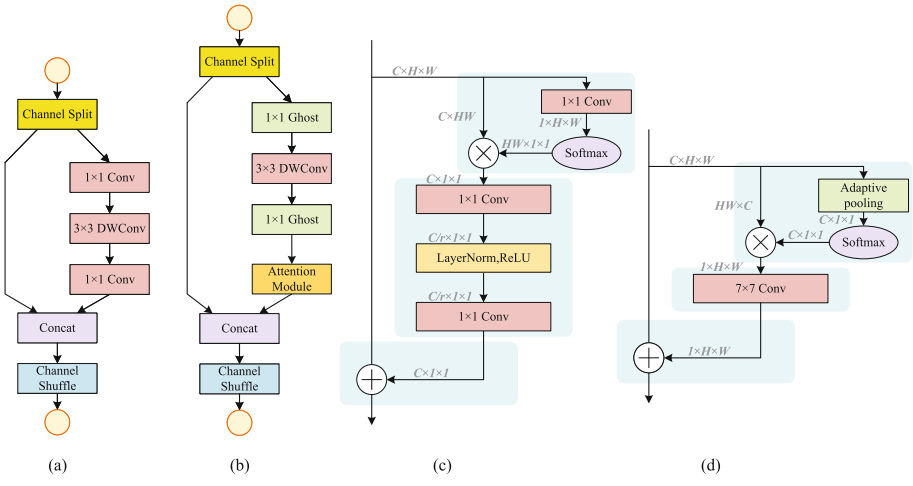

Fig. 2. (a) Shuffle block (b) efficient shuffle block (c) GC block (d) CGC block.

shows, this paper introduces the Ghost module proposed in GhostNet [6] to replace the costly pointwise (1×1) convolutions in shuffle blocks, which can reduce the model complexity while improving the performance.

GhostNet asserts that a handful of intrinsic feature-maps contain most of the dominant information of the output, so as shown in Fig. 3, an ordinary convolution is divided into two parts, a primary convolution for generating intrinsic feature-maps and a cheap operation for generating the remaining feature-maps.

Specifically, given the input data $X \in \mathbb{R}^{c \times h \times w}$, the desired output feature-maps $Y \in \mathbb{R}^{h' \times w' \times n}$, where c and n represent the input and output channel numbers, and h, w and h', w' represent the height and width of the input and output, respectively. A primary convolution is used to generate l intrinsic feature-maps $Y' \in \mathbb{R}^{h' \times w' \times l}$:

$$Y' = X * f' \tag{1}$$

where $f' \in \mathbb{R}^{c \times k \times k \times l}$ is the convolution filter with $k \times k$ kernel size, and $l \leq$ n.

Subsequently, a series of cheap operations are applied on each intrinsic feature-map in Y' to obtain the other $n - l$ feature-maps according to the following function:

$$y_{i,j} = \Phi_{i,j}\left(y_i'\right), \quad \forall i = 1, \ldots, l, \quad j = 1, \ldots, r-1 \tag{2}$$

where $y_i{}'$ is the i-th intrinsic feature-map in Y', $\Phi_{i,j}$ is the j-th linear operation on $y_i{}'$ for generating the feature-map $y_{i,j}$. Concatenating the output feature-maps of primary convolution and cheap operation, the final output of a Ghost module $Y = [[y_1, \ldots, y_l], [y_{1,1}, y_{1,2}, \ldots, y_{l,r-1}]]$ can be obtained, where $n = l \times 1 + l \times (r-1)$, r represents the proportion of intrinsic feature-maps, and l consequently equals to $\frac{n}{r}$.

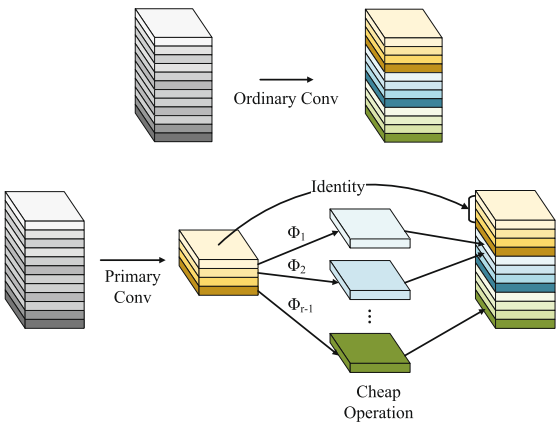

Fig. 3. The illustrations of a ordinary convolutional and a Ghost module.

In this paper, the kernel size of the Ghost module is set to 1 ($k = 1$), which results in a 1×1 Ghost module. In addition, r is set to 2, and following GhostNet, 3×3 convolutions are used as the cheap operations. Consequently, after

replacing the original 1×1 convolution with the 1×1 Ghost module, the theoretical compression ratio of parameters and speed-up ratio of computation can be calculated as:

$$\begin{aligned} R &= \frac{c \cdot k \cdot k \cdot n}{c \cdot k \cdot k \cdot \frac{n}{r} + (r-1) \cdot \frac{n}{r} \cdot d \cdot d} = \frac{c \cdot 1 \cdot 1 \cdot c}{c \cdot 1 \cdot 1 \cdot \frac{c}{2} + (2-1) \cdot \frac{c}{2} \cdot 3 \cdot 3} \\ &= \frac{2 \cdot c^2}{c^2 + 9 \cdot c} \approx 2 \end{aligned} \tag{3}$$

As the above formulation shows, computation and parameters have been halved.

Attention Module. Capturing long-range dependencies is critical for HPE, but simply stacking ordinary convolution layers cannot effectively extract the global understanding of the visual scene. Cao *et al.* [2] proposed a global context network (GCNet), in which the global context (GC) block can effectively aggregate the global information. Consequently, this paper introduces GC block to strengthen the capture ability of long-range dependencies. Figure 2 (c) depicts the structure of the GC block, which can be abstracted into three procedures: (a) global attention pooling, which employs a 1×1 convolution and a softmax function to obtain the attention weights, and then use these attention weights to perform the attention pooling, which aggregates the features of all positions together to acquire the global context features; (b) feature transform, which adopts 1×1 convolution to capture channel-wise interdependencies; (c) Feature aggregation, which uses element-wise addition to merge the global context features into all positions.

Meanwhile, inspired by CBAM [25], this paper redesigns the GC block and proposes a new attention block named channel global context(CGC) block, which emphasizes the modeling of inter-channel relationships. The GC block can be regarded as a kind of first-spatial-then-channel attention, in contrast to the GC block, the CGC block firstly aggregates the features of all channels together to form a global channel descriptor; and then models the inter-spatial relationships to obtain the final global channel context. Different modeling order leads to different attention maps, the CGC block focuses more on dependencies between channels. As shown in Fig. 2 (d), the CGC block can be summarized as three procedures: (a) Channel attention pooling, which firstly applies average-pooling along the spatial dimension and a softmax function to obtain the channel attention weights, then attention pooling is performed by matrix multiplication, and the features of all channels are aggregated together to obtain channel context features; (b) Spatial features transform, which adopts a 7×7 convolution to capture the inter-spatial relationships; (c) Feature aggregation, which employs element-wise addition for feature fusion.

The CGC block can act as a complementary for the GC block to compensate for its insufficient modeling ability of inter-channel relationships. This paper combines GC and CGC blocks to enhance the modeling ability of both inter-channel and inter-spatial relationships. Experimental results show that the

performance of GC-block-first combination form is slightly better than CGC-block-first form, so the attention module in the subsequent paper refers to the GC-block-first combination form.

3.2 Efficient Dense Connection

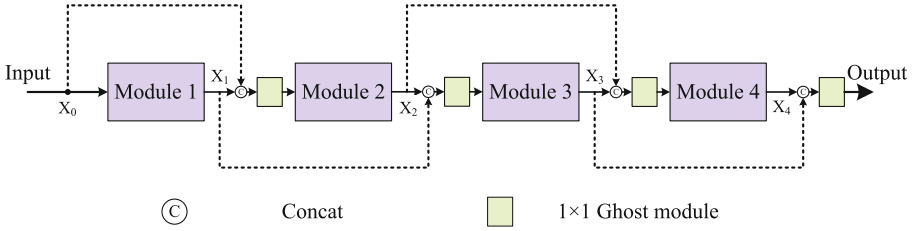

Fig. 4. The illustration of efficient dense connection.

DenseNet [13] proposed the concept of dense connection, which directly connects all layers with the same feature size to each other to ensure maximum information flow. Dense connections strengthen the feature propagation and reuse, alleviate vanishing-gradient problems, and make the network easy to train.

Inspired by DenseNet, this paper introduces dense connections between the different modules within the same stage and achieves feature reuse by concatenating operations. Existing studies [15] have shown that the connections between adjacent layers are much stronger than the others, therefore this paper only adds dense connections between the adjacent modules instead of all to achieve a better tradeoff between accuracy and speed.

Considering the s-th stage, there are s parallel branches with different resolutions. Each stage contains M modules, each of which implements a complex non-linear transformation $H_m(\cdot)$, where m is the index of the module. As the Module in Fig. 1 shows, $H_m(\cdot)$ contains a sequence of two efficient shuffle blocks and one multi-resolution fusion. This paper denotes the output of the m^{th} module as $\mathbf{x}_m$, the number of input channels as $\mathbf{c}_m$.

When the efficient dense connections are not added, the output of the m^{th} module is fed into the $(m+1)^{th}$ module as input, which can be expressed as:

$$\mathbf{x}_m = H_m\left(\mathbf{x}_{m-1}\right) \tag{4}$$

Figure 4 illustrates the layout of efficient dense connections. The m^{th} module receives the feature-maps of the preceding two modules (except $m = 1$), i.e., $\mathbf{x}_{m-2}$ and $\mathbf{x}_{m-1}$, as input:

$$\mathbf{x}_m = H_m\left(G_m\left(\left[\mathbf{x}_{m-2}, \mathbf{x}_{m-1}\right]\right)\right) \tag{5}$$

where $[\mathbf{x}_{m-2}, \mathbf{x}_{m-1}]$ is the concatenation of the feature-maps produced by modules $m-2$ and $m-1$. In addition, because the channel number has changed from $\mathbf{c}_m$ to $2\mathbf{c}_m$, this paper applies 1×1 Ghost module G_m to recover the channel number to match the m^{th} module. The 1×1 Ghost module adopts the same setting as Sect. 3.1.

This paper refers to the added connections as efficient dense connections because that, benefiting from the feature reuse of only adjacent modules and the efficiency of 1×1 Ghost module, these connections can facilitate convergence and promote network accuracy with a slight increase in model complexity.

4 Experiments

In this section, several experiments are conducted on two human pose estimation datasets, COCO [14] and MPII [1]. Comparative experiments between EHR-HPE and some state-of-the-art methods are conducted on both datasets, ablation studies are only carried out on MPII to demonstrate the effectiveness of each component.

4.1 Experimental Setup

Datasets. The COCO dataset [14] contains over 200K images and 250K person instances labeled with 17 keypoints. COCO is divided into train, validation, and test sets. This paper trains EHR-HPE on train2017 dataset, including 57K images and 150K person instances. Evaluations are carried out on val2017 set and test-dev2017 set, containing 5K images and 20K images, respectively. The MPII dataset [1] provides around 25K images containing over 40K labeled person instances, in which 12K instances are used for testing, and 28K are used for training.

Training. The network is implemented by PyTorch and random parameter initialization is used. Adam optimizer is adopted with a mini-batch of size 32 and 210 epochs are trained. The initial learning rate is set to $5e^{-4}$ and reduced by a factor of 10 at the 170th and 200th epoch. In data preprocessing, the human detection box is expanded to a fixed aspect ratio of 4: 3, and then crop the box from the images. The input images are resizeed to 384×288 for COCO dataset and 256×256 for MPII dataset. Data augmentation operations are performed on two datasets to strengthen models' robustness, including random rotation ($[-30°, 30°]$), random scale ($[0.75, 1.25]$), and random flipping, what's more, half body data augmentation is also used for COCO. All experiments are conducted on two NVIDIA 1080Ti GPUs.

Testing. For COCO, the two-stage top-down paradigm is used, i.e., firstly detect the person instance via a person detector provided by SimpleBaseline [26], and then predict keypoints. For MPII, the standard strategy (using the provided

person boxes) is adopted to guarantee the fairness of the results. Following the common practice [28,30], heatmaps are computed via averaging the heatmaps of the original and flipped images.

Evaluation Metrics. For COCO, this paper uses the OKS-based mAP metric and reports standard average precision and recall scores. OKS (Object Keypoint Similarity) represents the similarity between human poses. AP^{50} represents the AP scores at OKS = 0.50, AP^{75} represents the AP scores at OKS = 0.75, AP represents the mean of AP scores at 10 positions, OKS = 0.50, 0.55, ..., 0.95. AP^{M} represents AP for medium objects, AP^{L} represents AP for large objects. For MPII, this paper uses the standard metric PCKh@0.5 (detected keypoint is considered correct if the distance between the predicted and ground-truth keypoints is less than 50% of the length of head bone link) to evaluate the performance.

4.2 Results

Results on COCO Val. Table 1 gives the results of EHR-HPE and other state-of-the-art methods. The proposed EHR-HPE achieves 71.2 AP score when the input size is 384×288, with only 1.7M parameters and 0.91 GFLOPs. EHR-HPE outperforms Small HRNet-16[2] over 15 AP points. Compared to ShuffleNetV2 and MobileNetV2, EHR-HPE achieves 7.6 and 3.9 points gain, respectively, while taking on much lower complexity. Compared to LPN, EHR-HPE improves AP by 2.1 points with lower complexity. Lite-HRNet is an effective lightweight pose network, and EHR-HPE achieves better accuracy than it with a slight increase in computation cost. Compared to FLPN whose accuracy is only 0.1 points higher than ours, our parameter size is only 16.8% of it, and the computation cost is also lower. In comparison with large models, EHR-HPE achieves a better AP score than CPN, Hourglass, and SimpleBaseline, with much less complexity.

Due to the effectiveness of our efficient shuffle blocks and the feature reuse of efficient dense connections, the proposed EHR-HPE achieves a great tradeoff between accuracy and model complexity.

Results on COCO Test-Dev. Table 2 reports the results of EHR-HPE and the existing state-of-the-art methods. The proposed EHR-HPE achieves 70.1 AP score, which outperforms all the small networks. Compared to Lite-HRNet, EHR-HPE achieves 0.4 points gain with a little increase in computation cost. In comparison with large models, EHR-HPE outperforms Mask-RCNN, G-RMI and Integral Pose Regression, achieves acceptable results, and is much more efficient in terms of model size (Params) and computation cost.

Results on MPII Val. This paper evaluates EHR-HPE on MPII to further compare it with other lightweight networks and the results is shown in Table 3.

[2] Available from https://github.com/HRNet/HRNet-Semantic-Segmentation.

Table 1. Comparison on the COCO val set.

Model	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Large networks										
8-stage Hourglass [16]	8-stage Hourglass	256 × 192	25.1M	14.3	66.9	-	-	-	-	-
CPN [3]	ResNet-50	256 × 192	27.0M	6.20	68.6	-	-	-	-	-
SimpleBaseline [26]	ResNet-50	256 × 192	34.0M	8.90	70.4	88.6	78.3	67.1	77.2	76.3
HRNetV1 [21]	HRNetV1-W32	256 × 192	28.5M	7.10	73.4	89.5	80.7	70.2	80.1	78.9
DARK [29]	HRNetV1-W48	128 × 96	63.6M	3.6	71.9	89.1	79.6	69.2	78.0	77.9
Small networks										
Small HRNet	HRNet-W16	384 × 288	1.3M	1.21	56.0	83.8	63.0	52.4	62.6	62.6
ShuffleNetV2 1× [15]	ShuffleNetV2	384 × 288	7.6M	2.87	63.6	86.5	70.5	59.5	70.7	69.7
MobileNetV2 1× [10]	MobileNetV2	384 × 288	9.6M	3.33	67.3	87.9	74.3	62.8	74.7	72.9
LPN [30]	ResNet-50	256 × 192	2.9M	1.0	69.1	88.1	76.6	65.9	75.7	74.9
Lite-HRNet [28]	Lite-HRNet-30	384 × 288	1.8M	0.70	70.4	88.7	77.7	67.5	76.3	76.2
FLPN [20]	SResNet-50	256 × 192	10.0M	1.10	71.3	91.6	79.0	68.8	75.3	74.5
EHR-HPE	EHRNet	384 × 288	1.7M	0.91	71.2	89.1	78.8	69.0	76.8	75.3

The proposed EHR-HPE achieves 87.3 PCKh@0.5, outperforms Small HRNet-16, ShuffleNetV2, MobileNetV2, MobileNetV3 [9] and Lite-HRNet by 7.1, 4.5, 1.9, 3.0 and 0.3 points, respectively. Compared to FLPN, there is a little gap (0.5 points). However, the amount of the parameters and the computation cost of EHR-HPE are only 17% and 48% of FPLN, respectively.

Table 2. Comparision on the COCO test-dev set.

Model	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Large networks										
Mask-RCNN [7]	ResNet-50-FPN	-	-	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [17]	ResNet-101	353 × 257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [22]	ResNet-101	256 × 256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	-
CPN [3]	ResNet-Inception	384 × 288	-	-	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [4]	PyraNet	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	-
SimpleBaseline [26]	ResNet-152	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNetV1 [21]	HRNetV1-W32	384 × 288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNetV1 [21]	HRNetV1-W48	384 × 288	63.6M	32.9	75.5	92.5	83.3	71.9	81.5	80.5
DARK [29]	HRNetV1-W48	384 × 288	63.6M	32.9	76.2	92.5	83.6	72.5	82.4	81.1
Small networks										
Small HRNet	HRNet-W16	384 × 288	1.3M	1.21	55.2	85.8	61.4	51.7	61.2	61.5
ShuffleNetV2 1× [15]	ShuffleNetV2	384 × 288	7.6M	2.87	62.9	88.5	69.4	58.9	69.3	68.9
MobileNetV2 1× [10]	MobileNetV2	384 × 288	9.8M	3.33	66.8	90.0	74.0	62.6	73.3	72.3
LPN [30]	ResNet-50	256 × 192	2.9M	1.0	68.7	90.2	76.9	65.9	74.3	74.5
FLPN [20]	SResNet-50	256 × 192	10.0M	1.10	68.7	90.6	77.2	65.9	74.0	74.5
Lite-HRNet [28]	Lite-HRNet-30	384 × 288	1.8M	0.70	69.7	90.7	77.5	66.9	75.0	75.4
EHR-HPE	EHRNet	384 × 288	1.7M	0.91	70.1	91.2	77.9	66.7	75.6	75.2

4.3 Ablation Study

A series of ablation studies on MPII validation set have been conducted to analyze the effectiveness of each component proposed in this paper. Table 4 reports the results. Firstly, this paper combines shuffle blocks with Small HRNet as baseline, which achieves 86.38 points (PCKh@0.5) with 1.3 M parameters and 419 MFLOPs. To further improve the performance, two 1×1 Ghost modules (r is set to 2) are introduced to replace the two 1×1 convolutions in shuffle blocks, thereby the number of parameters and the computation cost are reduced by 0.17M and 60 MFLOPs, respectively, whereas the performance is improved by 0.13 points.

Table 3. Comparisons on the MPII val set.

Model	#Params	GFLOPs	PCKh@0.5
Small HRNet	1.3M	0.74	80.2
ShuffleNetV2 1×	7.6M	1.70	82.8
MobileNetV3 1×	8.7M	1.82	84.3
MobileNetV2 1×	9.6M	1.97	85.4
Lite-HRNet-30	1.8M	0.42	87.0
FLPN	10.0M	1.10	87.8
EHR-HPE	1.7M	0.53	87.3

Table 4. Ablation studies on the MPII val set.

Model	#Params	MFLOPs	PCKh@0.5
Small HRNet	1.3M	736	80.2
Baseline	1.3M	419	86.38
+ 1×1 Ghost	1.13M	359	86.51
+ 1×1 Ghost + GC block	1.24M	361	86.63
+ 1×1 Ghost + CGC block	1.13M	365	86.58
+ 1×1 Ghost + attention module	1.25M	368	86.69
EHR-HPE	1.69M	534	87.33

Subsequently, GC blocks are introduced into the shuffle blocks to strengthen the modeling ability on spatial long-range dependencies, and achieves 86.69 points. Based on GC block, CGC block is proposed to enhance the information exchange across channels, and achieves 86.63 points. This paper combines GC and CGC blocks as the attention module used in EHR-HPE to enhance the modeling ability of both inter-channel and inter-spatial relationships, performance has been improved by 0.18 points at the cost of 0.12M extra parameters and 9 extra MFLOPs.

Finally, efficient dense connections are proposed to facilitate convergence and promote network accuracy, resulting in the final EHR-HPE, which achieves 87.33 points with 1.69M parameters and 534 MFLOPs on MPII validation set.

5 Conclusion

Considering the deployment difficulty of large human pose estimation methods, this paper proposes an efficient and lightweight network named EHR-HPE. EHR-HPE network first follows the high-resolution pattern of Small HRNet to acquire accurate heatmaps; then an efficient shuffle block is designed to reduce the model complexity and boost model performance; finally, the efficient dense connections are added to further improve model accuracy. Extensive experiment results demonstrate that the proposed EHR-HPE network can achieve comparable results with those top-performing methods, while the model complexity is much lower, making it more suitable for resource-limited devices.

Acknowledgements. This work was funded by the Project Research on human-robot interactive sampling robots with safety, autonomy, and intelligent operations supported by NSFC (92048205).

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, pp. 3686–3693 (2014)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: GCNet: non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
3. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7103–7112 (2018)
4. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: RMPE: regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2334–2343 (2017)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Comput. Vision **61**(1), 55–79 (2005)
6. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: GhostNet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1580–1589 (2020)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Howard, A., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)

10. Howard, A., Zhmoginov, A., Chen, L.C., Sandler, M., Zhu, M.: Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation (2018)
11. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: exploiting feature context in convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
14. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
15. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 122–138. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_8
16. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
17. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4903–4911 (2017)
18. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595 (2013)
19. Qin, X., Guo, H., He, C., Zhang, X.: Lightweight human pose estimation: CVC-Net. Multimedia Tools Appl. **81**(13), 17615–17637 (2022)
20. Ren, H., Wang, W., Zhang, K., Wei, D., Gao, Y., Sun, Y.: Fast and lightweight human pose estimation. IEEE Access **9**, 49576–49589 (2021)
21. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)
22. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11210, pp. 536–553. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_33
23. Wang, X., Li, Z., Chen, Y., Jiang, P., Wang, F.: Stacked mixed-scale networks for human pose estimation. In: Nayak, A.C., Sharma, A. (eds.) PRICAI 2019. LNCS (LNAI), vol. 11670, pp. 217–229. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29908-8_18
24. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
25. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1

26. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11210, pp. 472–487. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_29
27. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR 2011, pp. 1385–1392. IEEE (2011)
28. Yu, C., et al.: Lite-HRNET: a lightweight high-resolution network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10440–10450 (2021)
29. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7093–7102 (2020)
30. Zhang, Z., Tang, J., Wu, G.: Simple and lightweight human pose estimation. arXiv preprint arXiv:1911.10346 (2019)