Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Lightweight and Effective Human Pose Estimation Model Based on Multi-Angle Knowledge Distillation

To cite this article: Hua Li 2022 *J. Phys.: Conf. Ser.* **2224** 012025

View the article online for updates and enhancements.

You may also like

- 3D human pose estimation based on negative exponential reduction Gaussian kernel
Lanqing Gu and Yu Wang
- High-Resolution Representation Learning for Human Pose Estimation based on Transformer
Dengyu Fu and Wei Wu
- A survey on Pose Estimation using Deep Convolutional Neural Networks
Manisha Patel and Nilesh Kalani

This content was downloaded from IP address 103.178.55.206 on 30/03/2023 at 09:04

Lightweight and Effective Human Pose Estimation Model Based on Multi-Angle Knowledge Distillation

Hua Li

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

Email: lihua_programmer@163.com

Abstract. In the field of human pose estimation, most of the existing methods focus on improving the generalization performance of the model, while ignoring the significant efficiency issues. This leads to an increasing amount of model parameters and needs to take up more and more computing resources, which greatly reduces the practical value of the model. In order to solve this problem, we propose a novel lightweight network structure called Effective and Lightweight Pose Network (ELPN). At the same time, for the sake of alleviating the difficulty of lightweight model training, we propose a Multi-Angle Pose Distillation (MAPD) model training method that can more effectively train particularly small pose network models. In quantitative experiments, our models have excellent performance on two mainstream benchmark datasets: the MPII and the COCO. In qualitative testing, our models can accurately locate the keypoints of complex human movements. These fully demonstrates the efficiency and effectiveness of our methods. Our models have the characteristics of high precision, small size and fast inference speed. It is a cost-effective model with greater practical value.

1. Introduction

Human pose estimation (HPE) is still an increasingly active research field of computer vision, and has broad application prospects, such as scene understanding [1], video surveillance [2], human action recognition [3, 4] and human computer interaction [5]. Its task is to locate the joints of human body parts, such as shoulders, wrists and knees.

Recently, deep learning has been widely used in the field of human pose estimation, especially the methods based on deep convolution neural network [6-14], which greatly promotes the development of this field. However, most of the current models improve performance at the cost of deployment overhead. These models usually have a large number of parameters and floating-point operations (FLOPs). As a result, they are particularly time-consuming when inferring, and also require the inference device to have a considerable amount of memory. Therefore, these state-of-the-art models cannot be directly deployed on resource-constrained devices such as tx2, smartphones and robots.

In our work, we design human pose estimation models with low parameters and computation complexity (FLOPs). At the same time, we use a novel model training method which is Multi-Angle Knowledge Distillation (MAKD) to obtain models with high performance. High-Resolution Network (HRNet) [6] is the state-of-the-art network model at present. It creatively designs a network architecture that can not only maintain high-resolution representation, but also continuously fuse features. However, like most existing models, it is not suitable for direct deployment on resource constrained devices. So we design a novel lightweight model based on HRNet called Effective and Lightweight Pose Network (ELPN). The minimum parameter of this model is only 1.1M, which is 3%

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

of HRNet model in the same situation. The lowest computational complexity (FLOPs) is only 1.6G, which is 22% of the HRNet model in the same case.

For the human pose estimation network, a pre-training model is usually trained on the ImageNet dataset, and then the pre-training model is used to train on the target dataset such as COCO dataset [15]. There are two main benefits of this: First of all, related experiments [6, 16] have shown that the performance of the network trained using the pre-training model is much better than that of the network trained from scratch. Secondly, the convergence speed of training using the pre-trained model is faster. But the ImageNet dataset is quite large, and training a pre-trained model on it requires a lot of computing resources and also consumes a lot of time. To overcome this problem, Zhang et al. [7] used knowledge distillation to train the target model. Lightweight Pose Network (LPN) [9] proposed an iterative training strategy to train the target model. The training method of knowledge distillation can not only improve the performance of the model, but also accelerate the convergence speed of the model. The former is obviously better than the latter. Inspire by Zhang et al. [7], we propose a novel method of knowledge distillation called Multi-Angle Pose Distillation (MAPD) to train lightweight network models. Different from FPD [7], our teacher network supervises and trains student networks from multiple angles, thus further improving the generalization ability of student network. The model training framework is shown in Figure 1.

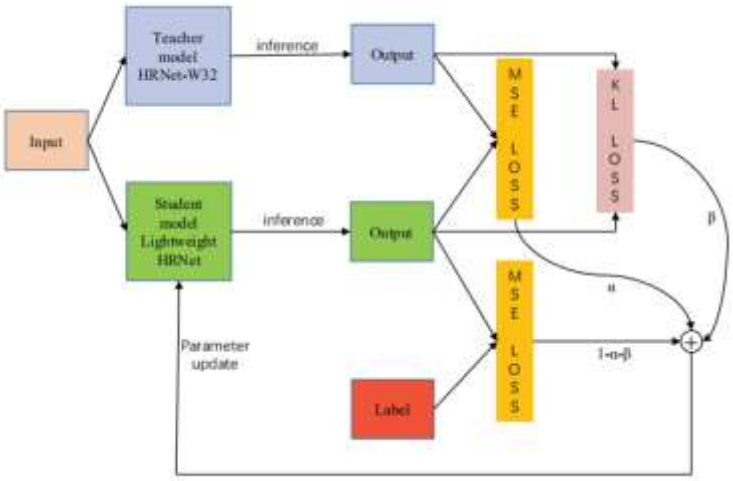

Figure 1. Overview of our model training framework. We use HRNet-W32 as teacher model and lightweight architecture as student model. The two models input the same data, and then get the output by inference, then calculate the MSE loss and KL loss respectively, using α and β as the weight parameters. In addition, the output of student network also needs to calculate MSE loss with real labels. Finally, the loss of the three parts is added by a certain weight as the total loss, which is used to update the parameters of student network model by back propagation.

We empirically demonstrate the efficiency and effectiveness of our methods on two mainstream datasets: the MPII Human Pose [17] and the MS-COCO [15]. The experiments show that our ELPN model has less parameters and lower computational complexity than other top-performing models, and can maintain high performance. It is a more cost-effective and practical model, which makes it possible to estimate human pose in real-time on edge devices.

In summary, our contributions are three-fold as follows:

(1) We systematically investigate the efficiency of human pose estimation models that are currently under-researched, and propose a novel lightweight network structure called Effective and Lightweight Pose Network (ELPN).

(2) We propose a Multi-Angle Pose Distillation (MAPD) model training method that can more effectively train particularly small pose network models. This training method has many advantages, such as fast convergence speed of model training, more stable training process and so on.

(3) Our models have excellent performance on the MPII and COCO datasets, which fully demonstrates the efficiency and effectiveness of our model. Our models have the characteristics of high precision, small size and fast inference speed. It has greater practical value.

2. Related Work

Human Pose Estimation. The traditional methods of studying human pose estimation problem is to use probabilistic graphical model [18] or pictorial structure model [19, 20]. With the rise of convolutional neural networks (CNN), researchers generally adopt methods based on CNN to study this problem. Prior works [6, 12, 16, 21-24] has indeed proved that using CNN-based methods is better than traditional methods. At present, most of the works regard human pose estimation as a regression problem. There are two mainstream ways to get the final keypoints: direct regression of keypoints coordinates [25] and regression of heatmaps [6, 13, 26], respectively using the maximum corresponding position of each heatmap as the keypoints coordinates.

Toshev et al. [25] proposed the DeepPose model, which is the first attempt to use CNN to directly regress the location of keypoints. Subsequently, more and more researchers use CNN to solve the problem of human pose estimation, which has led to a significant improvement in this field. For example, Cao et al. [22] proposed the OpenPose model, which used multi-stage heatmaps supervision to locate keypoints. Cascaded Pyramid Network (CPN) [12] used multi-scale feature fusion to help locate the keypoints of some small targets, and won the championship in the COCO keypoints challenge in 2017. Sun et al. [6] introduced a High-Resolution Network (HRNet), which can maintain high-resolution representation in the whole process, and then realize multi-scale feature fusion by repeatedly stacking exchange units. It's the best performing model so far. The predecessors paid more attention to the performance of the model, but ignored the cost of deploying the model on resource constrained devices, such as model parameter quantity and calculation cost. They usually use more complex network structure in order to improve a little bit of accuracy, such a model is not the most cost-effective, because they need to occupy a lot of memory and more computing resources, which makes it difficult to get applications in real life.

Lightweight Pose Estimation. In order to make the human pose estimation model more practical, researchers are also eager to study models with low parameters, computation overhead, and high precision. Representative works are [7, 9, 8]. Zhang et al. [7] proposed a training method called Fast Pose Distillation (FPD), using the Hourglass [14] network as teacher to train a lightweight student network. Although the network performance of the trained students has not descended, the calculation amount of the model has not been substantially reduced, and the model inference is still quite time-consuming. Bulat et al. [8] binarized the network model in an attempt to compress the model parameters while speeding up the model inference time, but they ignore the model performance issue. Finally, the performance of the model is obviously reduced, which can not meet the practical requirements. Lightweight Pose Network (LPN) [9] was based on SimpleBaseline [16], used lightweight convolution module to construct network model, and proposed an iterative training strategy to train the model, which ultimately reduced the amount of model parameters. However, the computational complexity of the model is still relatively high and the accuracy of the model still has room for improvement. Following the design principles of HRNet [6], we explore an efficient and lightweight network structure, which has the characteristics of small model, high precision and fast inference speed. In the experimental part, we will fully verify the effectiveness and efficiency of the model.

Attention Mechanism. In recent years, since the effectiveness of the attention mechanism has been extensively studied by researchers, it has also been widely used in the field of human pose estimation [27-32]. Chu et al. [28] introduced the attention mechanism into the field of human pose estimation for the first time, and its outstanding performance impressed us. Su et al. [29] used both spatial and

channel attention mechanisms to enhance the performance of the model, but this introduced a large parameter and computational burden. In order to meet the needs of lightweight, Cao et al. [31] proposed the Global Context (GC) block, which is a self-attention module that can capture long-range dependencies. It is improved on the basis of Non-local (NL) block [32], achieving the goal of smaller parameters and better performance. Therefore, in constructing our lightweight network model, we apply the GC block to improve the generalization ability of the model, which will not bring too many additional parameters and computational burden.

3. Approach

we design a novel lightweight variant of HRNet [6] model called Effective and Lightweight Pose Network (ELPN), which uses the deep separable convolution [33], Global Context (GC) [31] block and so on. We set up two versions of the ELPN network, denoted by ELPN-V1 and ELPN-V2. ELPN-V2 performs better than ELPN-V1, but the number of parameters and calculations are higher. At the same time, the Multi-Angle Pose Distillation (MAPD) model training method is proposed to train our ELPN models. Before formally introducing our work, let us briefly review the HRNet model.

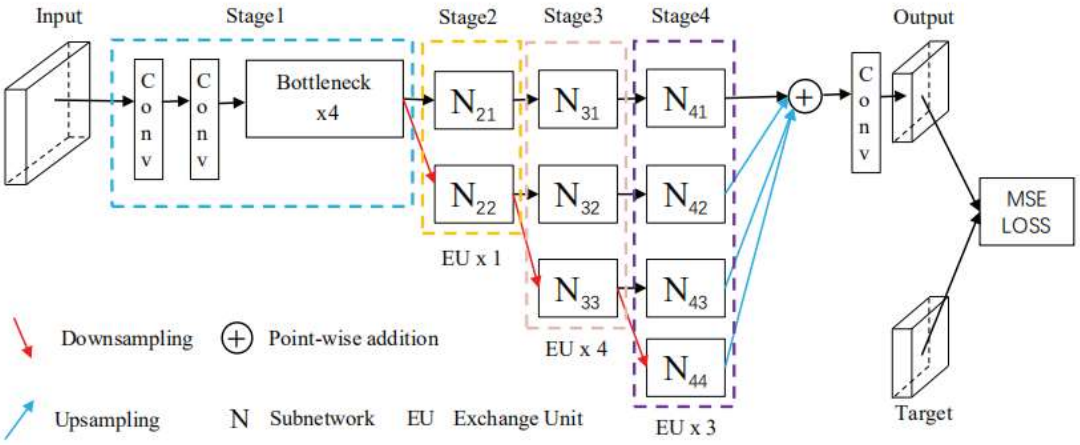

Figure 2. Architecture of High-Resolution Network. The network consists of four stages. The first stage is used to extract primary features, and from the second stage to the fourth stage, parallel subnetworks are added to extract higher-order features. Finally, the features extracted from all subnetworks are fused and the output is obtained by convolution. × represents the number of times the module is repeated.

3.1 High-Resolution Network

The biggest characteristic of the High-Resolution Network (HRNet) is that the network always maintains high resolution. Most of the previous CNN models reduce the resolution of the feature maps first, and then restore its resolution. In this process, the feature information will be lost. The HRNet will not lose the feature information, and the performance will be better naturally. The HRNet structure is shown in Figure 2. It mainly includes four stages. The first stage consists of two standard convolution layers and four standard bottleneck modules in ResNet [27]. From the second stage to the fourth stage, each stage contains two, three and four parallel subnetworks. N_{sr} is used to represent the subnetwork, s is the stage index and r is the resolution index. When r is increased by 1, the resolution of the feature maps is reduced by half and the number of channels is doubled. Each subnetwork is composed of four standard Basic Blocks in ResNet [27]. All parallel subnetworks in the same stage form an Exchange Unit (EU) through feature fusion. Take the third stage as an example. Its EU structure is shown in Figure 3. From the second stage to the fourth stage, there are 1, 4, and 3

Exchange Units (EUs). Finally, the feature maps obtained in the fourth stage are fused by point-wise addition, and the output heatmaps are obtained by using a single convolution layer.

In the task of human pose estimation, Sun et al. [6] instantiated two versions of the network structure: HRNet-32 and HRNet-48. 32 and 48 represent the number of channels of the feature maps when the resolution index r is 1. So when r is 2, 3, 4, the corresponding channel numbers are 64, 128, 256 and 96, 192, 384, respectively. Our research is mainly conducted on HRNet-32.

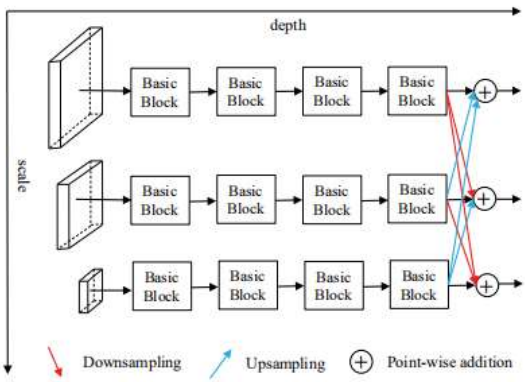

Figure 3. Illustration about the structure of the Exchange Unit (EU). Take an EU in the third stage as an example. It contains 3 parallel sub-networks, the resolution is gradually reduced from top to bottom, and the number of channels is continuously increasing. Each subnetwork is composed of four standard Basic Blocks in ResNet [27]. Finally, the features of each subnetwork are fused.

3.2 Compact Pose Network Architecture

In this part, we will introduce the construction process of lightweight network model ELPN in details. We systematically investigate the network structure of HRNet [6], and through the corresponding experiments, we find that the number of channels and the number of Exchange Units (EUs) in each stage have a great impact on the parameters and floating-point operations of the model. At the same time, HRNet [6] is composed of a large number of standard Bottleneck and Basic blocks in ResNet [27]. This inspired us to transform the lightweight Bottleneck and Basic blocks.

First of all, we conducted an experiment on the influence of the number of feature map channels on the generalization ability of the model. We keep the other parameters are same and only change the channels number of the feature map. Experiment is carried out when the initial number of channels is 8, 16, 32 and 64 respectively. The details of the experiment are shown in table 4. We are surprised to find that compared with 64 channels, 16 channels and 32 channels can maintain nearly 94% and 99% accuracy respectively on MPII dataset. However, compared with the 64 channels, the number of parameters and computation of them are reduced too much. This means that when the number of feature map channels for model initialization is set to 64, feature redundancy occurs, whereas when it is set to 8, insufficient feature expression occurs. To balance the lightweight and performance of the models, we designed two versions of the network, ELPN-V1 and ELPN-V2, with 16 and 32 initialization channels, respectively. So, the number of channels of other parallel subnetworks is 32, 64, 128 and 64, 128, 256 separately.

Subsequently, we analyze the impact of the number of EUs used at each stage on model parameters, computation overhead, and generalization capability. In the original HRNet model, from the second stage to the fourth stage, the number of EUs used is 1, 4, 3. Through experiments, we find that the appropriate reduction of the number of EUs used in the third and fourth stages has little effect on the generalization ability of the model, but can significantly reduce the model parameters and computation overhead. Especially when the number of EUs used is 1, 2, 2, the accuracy of the model has not decreased, on the contrary, it has been improved. At the same time, the parameter amount and computational cost of the model are significantly reduced. The specific experimental data are shown in

table V. To balance the lightweight and performance of the models, in our ELPN-V1 network, from the second stage to the fourth stage, we set the number of EUs used as 1, 2, 2. ELPN-V2 network is set to 1, 4, 3.

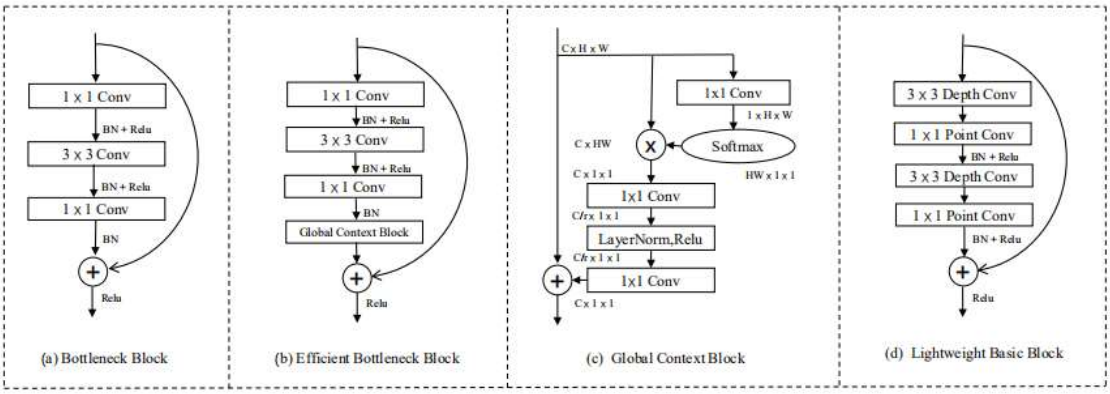

Figure 4. Illustrations about efficient and lightweight blocks. (a) In ResNet [27], the standard Bottleneck Block. (b) Adding the Global Context (GC) Block to the Bottleneck Block constitutes an Efficient Bottleneck Block. (c) The specific structure of Global Context Block. (d) Lightweight Basic Block structure, which is stacked by separable convolution.

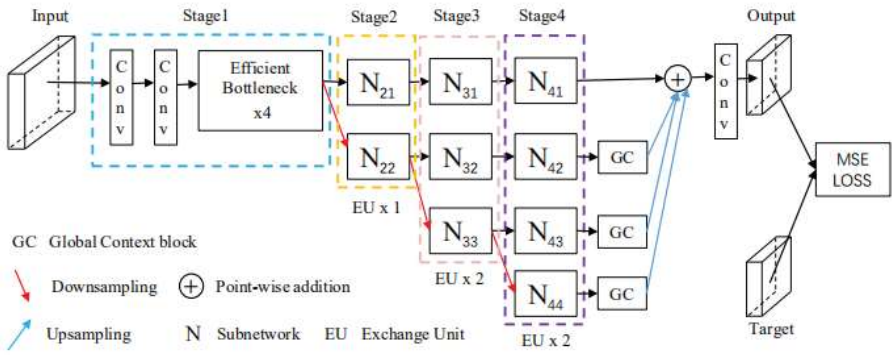

Figure 5. The framework of our Effective and Lightweight Pose Network (ELPN). We replace the original Bottleneck Block and the Basic Block in HRNet with Efficient Bottleneck Block and Lightweight Basic Block. Then add GC Block after the parallel sub-network in the fourth stage. At the same time, the number of Exchange Units from the second stage to the fourth stage is reduced to 1, 2, and 2, respectively, and the Effective Basic Block in the Exchange Unit is reduced from 4 to 2.

Finally, we modified the basic units of the models, the Bottleneck and Basic blocks, and named them Efficient Bottleneck Block and Lightweight Basic Block respectively. Bottleneck blocks are used four times in HRNet to extract low-level features. The Basic block is heavily reused in the second to fourth stages to extract high-level features. Based on this characteristic, we attempt to make the Bottlneck block more efficient and the Basic block lighter. The standard Bottleneck block is shown in Figure 4(a), which consists of three convolutional layers. We added the Global Context (GC) block [31] after its last convolutional layer to form the Efficient Bottleneck Block. Its structure is shown in Figure 4(b). The function of the GC block is to capture the long-range dependent features. Using it in the Bottleneck block can help us capture the contact information between different keypoints. The

principle of the GC block is shown in Figure 4(c). In order to further reduce the amount of parameters and calculations of the model, we replaced all the convolutions in the Basic block with separable convolutions [33], which are widely used in the design of compact network structures. Figure 4(d) shows the structure of the lightweight basic block. In order to avoid the problem that the network is too deep and difficult to train, we changed the number of lightweight Basic blocks in the EU from 4 to 2. In addition, we add a GC block after the feature maps output in the fourth stage to further capture the information between different keypoints. Our lightweight network structure is shown in Figure 5.

4. Supervision Enhancement by Multi-Angle Pose Distillation

For the task of human pose estimation, most scholars at present regard it as a regression problem, and finally use the MSE loss function as the supervision information, so that the heatmaps output by the model continuously approximates the label heatmaps. In our opinion, this task can also be regarded as a classification task. For a certain keypoints of a person in a given picture, we can divide the pixels into keypoints pixels and background pixels like semantic segmentation task. So we can use the classification loss function to guide the model to learn the distribution information of the heatmap. It is based on this idea that we propose a model training method of Multi-Angle Pose Distillation (MAPD) to train our lightweight models.

Our model's training framework is shown in Figure 1. We use HRNet-W32 as the teacher model. It is only responsible for forward inference and does not update model parameters. We directly use the model parameters trained by Sun et al. [6]. Use the lightweight variant of HRNet model as the student

model, which is the EPLN models of our design. The input data enters the teacher model and the student model at the same time, and then respectively output the corresponding heatmaps. We look at the problem of human pose estimation from the perspective of regression and classification respectively, and use the MSE loss function and KL loss function to let the teacher model guide the student model learning. At the same time, we also use the MSE loss function to monitor how the student network outputs fit the true label heatmaps.

We assume that L_{mpd} and L_{kpd} represent the loss function using MSE and KL as the pose distillation, respectively. The formula is expressed as:

$$\boldsymbol{L_{mpd}} = \frac{\mathbf{1}}{\boldsymbol{K}}\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\left\|\boldsymbol{o_k^s} - \boldsymbol{o_k^t}\right\|_{\mathbf{2}}^{\mathbf{2}} \tag{1}$$

$$\boldsymbol{L\_kpd}(\boldsymbol{o^t}\|\boldsymbol{o^s}) = \frac{\mathbf{1}}{\boldsymbol{K}}\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}}\boldsymbol{o_k^t}\left(\mathbf{log}\,\boldsymbol{o_k^t} - \mathbf{log}\,\boldsymbol{o_k^s}\right) \tag{2}$$

where $\boldsymbol{o_k^t}$ and $\boldsymbol{o_k^s}$ represent the output of the teacher and student model respectively. K is the number of heatmaps, and k is a certain heatmap of the output. We suppose that L_{mse} represents the loss function between the output of the student model and the label. Its expression is almost the same as Equation 1. Then the whole loss function L_{mapd} formula is as follows:

$$\boldsymbol{L_{mapd}} = \boldsymbol{\alpha L_{mpd}} + \boldsymbol{\beta L_{kpd}} + (\mathbf{1} - \boldsymbol{\alpha} - \boldsymbol{\beta})\boldsymbol{L_{mse}} \tag{3}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the weight coefficients used to control the pose distillation. We use Lmapd to guide students' network model learning, and constantly update model parameters until the model converges.

Generally speaking, it is a coarse localization method to treat the human pose estimation task as a classification problem, and then use KL loss function to make the distribution of students' network approximate to that of teachers'network. Its greatest advantage is that it can ensure that the output distribution of student network is consistent with that of teacher network. Regarding it as a regression

task, and then using MSE loss function to measure the output of student network and teacher network is a more accurate method, but this method can not perceive the distribution of output. When they are combined to guide the student model, the student model can not only learn the distribution information of the teacher network output, but also accurately locate the keypoints. This will promote students to learn better and learn more fully. In the experiment part, we will fully discuss the effectiveness of the training method.

5. Experiments

In this part, we will verify the effectiveness and efficiency of our models through quantitative and qualitative experiments. At the same time, the advantages of our training method are explained through corresponding ablation experiments. Our propose method is evaluated on two current mainstream human pose estimation datasets: COCO Keypoints Challenge dataset [15] and MPII Human Pose dataset [17].

6. COCO Keypoints Detection

Datasets and Evaluation metric. The COCO dataset is one of the most important datasets in the field of human pose estimation. Its train2017 dataset has 57k images and contains 150k human instances. Our model is only trained on the COCO train2017 dataset without using any additional data. The val2017 dataset contains 5k images and is mainly used to verify the effectiveness of the model offline. The test-dev2017 dataset contains 20k images. The final result will be compared with other public methods on this test-dev2017 dataset. All human body instances are labeled with 17 keypoints by default. For the COCO dataset, we use standard evaluation metric, which is based on the Object Keypoint Similarity (OKS) [15]. Its specific formula is as follows:

$$\mathbf{OKS} = \frac{\sum_i[\exp(-\mathbf{d_i^2}/2\mathbf{s^2}\mathbf{k_i^2})\delta(\mathbf{v_i}>0)]}{\sum_i[\delta(\mathbf{v_i}>0)]} \quad (4)$$

$\boldsymbol{d_i}$ represents the Euclidean distance between the keypoint coordinates of the ground-truth and the corresponding keypoint coordinates detected by the model. s means human scale, $\boldsymbol{k_i}$ represents a constant for controls falloff, and $\boldsymbol{v_i}$ reflects the visibility of real marked keypoints. There are three kinds of values of $\boldsymbol{v_i}$: 0, 1, 2, respectively, corresponding to the keypoint is not marked, the keypoint is marked but not visible in the image, and the keypoint is marked and visible in the image. We calculate Average Precision (AP) based on OKS to evaluate the accuracy of keypoint location.

Training. We use the proposed Multi-Angle Pose Distillation method to train our models. In the whole process, only the student network models need to be trained. When the training is completed, we directly discard the teacher model and use the student models for testing. Adam optimizer [34] and a minibatch of size 32 (Not fixed) are used to update the parameters. At the beginning, the learning rate is set to 1e-3, and then decreased by a factor of 0.1 at 170 and 200 epochs. Finally, a total of 210 epochs were trained.

We have the same data processing method as HRNet [6]. The human detection box is extended to a fixed aspect ratio (e.g., height: width = 4:3), then crop the box from the original picture, finally resize it to a fixed size (256 ×192) as the input image. The data augmentation includes random rotation (±40), flipping, and random scale (±30%). All our experiments are done with Pytorch on four NVIDIA 1080Ti GPU.

Testing. Our methods follow the top-down pipeline. Firstly, the human bounding boxes are detected in advance, and then the keypoints of each person are located according to it. For a fair comparison, we directly use the COCO val2017 and COCO test-dev2017 human bounding boxes provided by HRNet [6]. As the common practice, we calculate the average value of the original image and the flipped image heatmaps as the final heatmaps. Then, the position of the final keypoints are obtained by shifting the maximum response to the second largest response by a quarter.

Table 1. Comparison of results on the COCO val2017 dataset. The value of Pretrain is Y, which means the pre-training model is used, and N means that the pre-training model is not used.

Method	Backbone	Pretrain	Input Size	Params	FLOPs	AP	AP .5	AP .75	AP (M)	AP (L)	AR
8-stage Hourglass [14]	Hourglass	N	256 ×192	25.6M	26.2G	66.9	-	-	-	-	-
CPN [12]	ResNet-50	Y	256 ×192	27.0M	6.2G	68.6	-	-	-	-	-
LPN [9]	ResNet-50	N	256 ×192	2.9M	1.0G	69.1	88.1	76.6	65.9	75.7	74.9
LPN [9]	ResNet-101	N	256 ×192	5.3M	1.4G	70.4	88.6	78.1	67.2	77.2	76.2
SimpleBaseline [16]	ResNet-50	Y	256 ×192	34.0M	8.9G	70.4	88.6	78.3	67.1	77.2	76.3
LPN [9]	ResNet-152	N	256 ×192	7.4M	1.8G	71.0	89.2	78.6	67.8	77.7	76.8
SimpleBaseline [16]	ResNet-101	Y	256 ×192	53.0M	12.4G	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline [16]	ResNet-152	Y	256 ×192	68.6M	15.7G	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32 [6]	HRNet-W32	N	256 ×192	28.5M	7.1G	73.4	89.5	80.7	70.2	80.1	78.9
HRNet-W32 [6]	HRNet-W32	Y	256 ×192	28.5M	7.1G	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48 [6]	HRNet-W48	Y	256 ×192	63.6M	14.6G	75.1	90.6	82.2	71.5	81.8	80.4
SimpleBaseline [16]	ResNet-152	Y	384 ×288	68.6M	35.6G	74.3	89.6	81.1	70.5	79.7	79.7
HRNet-W48 [6]	HRNet-W48	Y	384 ×288	63.6M	32.9G	76.3	90.8	82.9	72.3	83.4	81.2
ELPN-V1(Ours)	HRNet-W16	N	256 ×192	1.1M	1.2G	70.5	89.3	78.0	67.6	76.2	75.7
ELPN-V2(Ours)	HRNet-W32	N	256 ×192	4.7M	2.0G	71.8	91.5	79.4	69.3	75.6	74.7

Results on COCO val2017. As shown in Table 1, we design two different versions of lightweight model (ELPN-V1, ELPN-V2). Only the number of initial feature map channels and exchange units (EUs) are different, and all other conditions are the same. The number of initial feature map channels of ELPN-V1 is 16, the number of EUs used in the second to fourth stages of the network is 1, 2, 2, and the corresponding parameters of ELPN-V2 are 32 and 1, 4, 3. Our lightest model ELPN-V1 achieves an AP score of 70.5% with only 1.1M parameters and 1.2G computational complexity. ELPN-V2 achieves a very competitive AP score of 71.8%.

Compared with hourglass [14] model, our lightest model ELPN-V1 has only 4% (1.1/25.6) of its parameters and 5% (1.2/26.2) of its computational complexity. However, the AP score increases by 3.6% (70.5-66.9). Compared with CPN [12], our models also keep the characteristics of low number of parameters, low computational complexity and high accuracy. The AP scores are increased by 1.9% (70.5-68.6) and 3.2% (71.8-68/6) respectively. LPN [9] is a typical lightweight model. Our model has more advantages than theirs. Although it increases a little computation, our model has less parameters and higher AP score. ELPN-V1 is 1.4% (70.5-69.1) higher than the LPN with Resnet-50 backbone, and ELPN-V2 is 0.8% (71.8-71.0) higher than the LPN with Resnet-101 backbone. Compared with the SimpleBaseline [16] and HRNet [6] models, although their AP scores are higher, their parameters and computation costs are much larger than ours. This will consume too much memory and computing resources, and the cost performance is not high. Our model is more cost-effective.

Table 2. Comparison of final results of COCO test-dev2017 dataset. We focus on comparing model parameters, floating point calculations (FLOPs) and average precision.

Method	Backbone	Input Size	Params	FLOPs	AP	AP .5	AP .75	AP (M)	AP (L)	AR
Mask-RCNN [21]	ResNet-50-FPN	-	-	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [35]	ResNet-101	353 ×257	42.6M	57.0G	64.9	85.5	71.3	62.3	70.0	69.7
Integral Regression [36]	ResNet-101	256 ×256	45.0M	11.0G	67.8	88.2	74.8	63.9	74.0	-
LPN [9]	ResNet-50	256 ×192	2.9M	1.0G	68.7	90.2	76.9	65.9	74.3	74.5
LPN [9]	ResNet-101	256 ×192	5.3M	1.4G	70.0	90.8	78.4	67.2	75.4	75.7
SimpleBaseline [16]	ResNet-50	256 ×192	34.0M	8.9G	70.0	90.9	77.9	66.8	75.8	75.6
LPN [9]	ResNet-152	256 ×192	7.4M	1.8G	70.4	91.0	78.9	67.7	76.0	76.2
SimpleBaseline [16]	ResNet-152	256 ×192	68.6M	15.7G	71.6	91.2	80.1	68.7	77.2	77.3
PNFS-1 [37]	MobileNet-v2	384 ×288	6.1M	4.0G	67.4	89.0	73.7	63.3	74.3	73.1
PNFS-2 [37]	ResNet-50	384 ×288	27.5M	11.4G	70.9	90.4	77.7	66.7	78.2	76.6
SimpleBaseline [16]	ResNet-152	384 ×288	68.6M	35.6G	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32 [6]	HRNet-W32	384 ×288	28.5M	16.0G	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48 [6]	HRNet-W48	384 ×288	63.6M	32.9G	75.5	92.5	83.3	71.9	81.5	80.5
ELPN-V1(Ours)	HRNet-W16	256 ×192	1.1M	1.2G	70.2	90.9	78.5	67.0	76.1	75.8
ELPN-V2(Ours)	HRNet-W32	256 ×192	4.7M	2.0G	71.4	91.1	79.3	68.1	77.5	77.0

Results on COCO test-dev2017. Table 2 illustrates the results of our models and methods in recent years on the COCO test-dev 2017. We mainly focus on parameter quantity, computational complexity and AP score index. As can be seen from the table 2, our models outperform Mask-RCNN [21], G-RMI [35], Integral Regression [36], LPN [9] and PNFS [37]. In particular, compared with PNFS [37] models using NAS method, our model has absolute advantages in parameters, computational complexity and AP score. Compared with the best performance model [6], ELPN-V1 only needs 1.7% (1.1/63.6) of the parameters and 3.6% (1.2/32.9) of the GLOPs, but gaining 93% (70.2/75.5) of the performance on the AP score. Similarly, ELPN-V2 only needs 7.4% (4.7/63.6) of the parameters and 6.1% (2.0/32.9) of the computation cost, and achieves 95% (71.4/75.5) of the AP score. These experimental data fully show that our models are more cost-effective than the recent ones. It can be deployed on edge devices at a lower cost to meet the diversified needs.

Table 3. Comparison of final results on the MPII test dataset. We focus on the model's performance in terms of total, parameter amount and computational cost.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	Params	FLOPs
Tompson et al. [38]	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6	-	-
Rafi et al. [39]	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3	56M	28G
Belagiannis et al. [40]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1	17M	95G
Insafutdinov et al. [41]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5	66M	286G
Wei et al. [42]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5	31M	351G
Bulat et al. [44]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7	76M	67G
Yang et al. [37]	97.9	95.6	90.7	86.5	89.8	86.0	81.5	90.2	6M	5G
Newell et al. [14]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9	26M	55G
zhang et al. [7]	98.3	96.4	91.5	87.4	90.9	87.1	83.7	91.1	3M	9G
Ning et al. [45]	98.1	96.3	92.2	87.8	90.6	87.6	82.7	91.2	74M	124G
Xiao et al. [16]	98.5	96.6	91.9	87.6	91.1	88.1	84.1	91.5	69M	21G
Tang et al. [43]	98.4	96.9	92.6	88.7	91.8	89.4	86.2	92.3	16M	34G
ELPN-V1(Ours)	96.5	94.2	86.9	83.2	88.1	82.3	79.6	87.6	1M	2G
ELPN-V2(Ours)	97.8	95.3	90.0	85.0	89.2	85.1	80.9	89.5	5M	3G

7. MPII Dataset

Datasets and Evaluation metric. The MPII dataset contains about 25k images, more than 40k human instances. Each human body instance can be labeled with 16 keypoints at most. The images in this dataset are obtained from YouTube human activities videos, which contain more than 410 activities. Like most existing methods, we randomly sampled 3k data from the training set as the valid set, and the remaining 22K as the training set. In the MPII dataset, we use the standard metric Percentage of Correct Keypoints with respect to head (PCKh) [17] to evaluate performance. The equation is expressed as follows:

$$\frac{\|y_i - \hat{y}_i\|_2}{\|y_{rhip} - y_{lsho}\|_2} \leq r \tag{5}$$

y_i and $\hat{y}_i$ represent the ith ground-truth keypoint coordinates and predicted keypoint coordinates respectively. y_{lsho} and y_{rhip} are the ground-truth coordinates of left shoulder and right hip respectively. r is a threshold value in the range of 0 to 1.$\|y_i - \hat{y}_i\|_2$ means the torso distance. In our results, PCKh@0.5 (r = 0.5) is used to report.

Training. Similar to experiments on COCO dataset, the Adam optimizer and minibatch size of 32 are used to update parameters. The initial learning rate, learning rate adjustment strategy and epoch of training are same as those in COCO dataset. The only difference is that the size of the input human bounding box is 256 × 256. The data augmentation includes flip, random scale 355 (±40%), random rotation (±30◦).

Testing. The method of testing is the same as operating on the COCO dataset. The only difference is that we use the standard testing strategy, which uses the human bounding boxes provided by the dataset itself instead of the human bounding box detected by the human detector.

Table 4. The ablation experiment on the influence of the number of feature map channels on the HRNet model accuracy, parameter amount and FLOPs in the initial stage.

Number of channels	mAP(PCKh@0.5)	Params	FLOPs
8	58.1	2.1M	1.8G
16	84.5	7.4M	3.4G
32	89.2	28.5M	9.5G
64	90.0	112.6M	33.4G

Results on MPII dataset. The PCKh@0.5 results on MPII test dataset are shown in Table 3. As shown in the table, our models achieve 87.6% and 89.5% PCKh@0.5 scores on the MPII test dataset, respectively. Compared with most models, our models have the characteristics of low parameter quantity, low computational complexity and high precision. Specifically, our models' parameters and calculation are lower than those in [38-42], but PCKh@0.5 score is higher than theirs. It is worth noting that the PCKh@0.5 score of Zhang et al's method [7] is slightly higher than ours, but his computational overhead is very large, and Yang et al's method [37] also has the same problem. Compared with the model with the highest PCKh@0.5 score [43], our ELPN-V1 model has only 6.3% (1/16) of its parameters and 5.9% (2/34) of its computation cost, but gaining 95.0% (87.6/92.3) performance in PCKh@0.5 accuracy. More importantly, our ELPN-V2 model has only 31.3% (5/16) of its parameters and 8.8% (3/34) of its computation, but maintains 97.0% (89.5/92.3) accuracy. This shows that our models are lightweight, effective and efficient, and have more advantages than the previous models.

4.3 Ablation Study

In this part, we will illustrate our process of constructing a lightweight network model through corresponding experiments, and will also verify the effectiveness of our proposed model training method. All experiments are performed on the MPII valid dataset. Except for special instructions, the default input resolution of the model is 256 × 256.

Effect of The Number of Channels: The default initialization channel number of the original HRNet [6] model is 32, which means that when the resolution index value of the model is 1, the number of channels in the feature maps is 32. The number of channels in the parallel sub-network feature maps is sequentially 64, 128 and 256, corresponding to resolution index values 2, 3 and 4. We first explored the influence of the number of channels on the performance of the model, the amount of parameters and the computational cost. Under the condition of ensuring that other parameters are the same, four sets of comparative experiments are set for the number of channels to be 8, 16, 32 and 64. The experimental results are shown in Table 4.

Table 5. Ablation study on the effects of the number of exchange units on the accuracy, parameters and flops of the HRNet network model from the second stage to the fourth stage.

Number of Exchange Units	mAP(PCKh@0.5)	Params	FLOPs
(1, 1, 1)	88.3	10.1M	4.4G
(1, 2, 1)	88.0	11.8M	5.3G
(1, 3, 1)	86.3	13.5M	6.2G
(1, 2, 2)	89.5	18.8M	6.5G
(1, 1, 3)	88.8	23.4M	6.8G
(1, 4, 3)	89.2	28.5M	9.5G

Effect of The Number of Exchange Units: The structure of the exchange units (EUs) is shown in Figure 3. The original HRNet [6] used repeated UEs to construct the network. The number of EUs reused from the second stage to the fourth stage is 1, 4, 3. In order to build our lightweight network model, we explored the influence of the number of EUs on the model generalization ability, parameter amount and computational overhead. In the experiment, we kept the initial number of channels at 32, other parameters were the same, and only changed the number of exchange units used in each stage of the model. A total of 6 sets of comparative experiments were carried out. The experimental results are shown in Table 5.

Table 6. Ablation experiment of knowledge distillation model training method. The meaning of ELPN is that our effective and lightweight pose network (ELPN) model is not trained using any knowledge distillation method. MSET and KLT respectively represent the knowledge distillation model training method using MSE and KL loss functions.

Method	mAP(PCKh@0.5)	Params	FLOPs
ELPN	78.2	1.1M	1.6G
ELPN+MSET	84.9	1.1M	1.6G
ELPN+KLT	83.6	1.1M	1.6G
ELPN+MSET+KLT	86.3	1.1M	1.6G

From the experimental results, the number of EUs has little influence on the generalization ability of the model, but has great influence on the model parameters and computational overhead. When the number of EUs is 1, 2, 2 and 1, 4, 3, the model has the best generalization ability. So, our ELPN-V1 small model uses the number of EUs of 1,2,2, and ELPN-V2 large model uses the number of EUs of 1,4,3. It should be emphasized that here we have selected the two sets of EUs parameters with the best generalization ability, because subsequent further lightening of the model may reduce the generalization ability of the model. Here, choosing the model with the best generalization ability can compensate for its subsequent reduction.

Effect of Knowledge Distillation Model Training Method: In this part, we use the already designed lightweight model ELPN-V1 to conduct experiments. It initializes the number of channels to 16, and the number of exchange units used from the second stage to the fourth stage is 1, 2, and 2. At the same time, it is composed of our improved Efficient Bottleneck Block and Lightweight Basic Block. We designed a total of four sets of experiments. They are not using the knowledge distillation method to train the model, using only the regression based MSE loss teacher network for knowledge distillation,

denoted as MSET, and only using the classification-based KL loss teacher network for knowledge distillation, denoted as KLT, and the training method used by both. We directly use HRNet-32 that has been trained in [6] as the teacher network. The experimental results are shown in Table 6.

It can be seen from the experimental results that both the regression-based model distillation method (MSET) and the classification-based model distillation method (KLT) can improve the generalization ability of the model, but the improvement effect is not as good as using both. When using MSET and KLT respectively, the generalization ability of the model has been improved by 6.7% (84.9-78.2) and 5.4% (83.6-78.2), which verifies that the model training method based on knowledge distillation is indeed effective. At the same time, the effect of MSET is better than KLT. When the two are used together, the generalization ability of the model has been improved by 8.1% (86.3-78.2). This shows that when two different perspectives of knowledge distillation model training methods are combined, a model with better generalization ability can be trained.

The reason why KLT alone is not as effective as MSET may be that KL loss is more inclined to perceive the overall distribution, and lack of grasp of details. And MSE loss can better perceive details, which is more advantageous for keypoint detection tasks. Combining the two can further enhance the generalization ability of the model, indicating that they can make up for each other's shortcomings, thus exerting their greatest potential.

Table 7. Ablation experiment on the weight parameter α that controls the loss of knowledge distillation.

α	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
mAP(PCKh@0.5)	78.2	85.3	85.5	86.3	85.7	85.5	85.9	85.2	85.1	85.2

Knowledge Distillation Weight Parameter Search: As shown in Equation 3, the final total loss is composed of MSE pose distillation loss, KL pose distillation loss, and MSE loss supervised by real landmark. Use $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as weight coefficients to control MSE pose distillation and KL pose distillation, respectively. This part of the experiment mainly explores the value of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. All experiments are based on ELPN-V1, except that the coefficients of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are different, all other conditions remain the same. It can be seen from Table 6 that the distillation effect based on MSE is better than that based on KL. For the sake of simplicity, we set the coefficient of $\boldsymbol{\alpha}$ to twice the $\boldsymbol{\beta}$, so we only need to search for the value of $\boldsymbol{\alpha}$. We set the value of $\boldsymbol{\alpha}$ with an initial value of 0, a step size of 0.05, and then end value of 0.45, and a total of 10 numbers are taken, which means that a total of 10 sets of experiments have been carried out.

The experimental results are shown in Table 7. When the $\boldsymbol{\alpha}$ is set to 0.15, the performance of the model is the best, which is 8.1% (86.3-78.2) higher than when the value is set to 0. This is a very big improvement. So, in the end we set the value of $\boldsymbol{\alpha}$ to 0.15 and the value of $\boldsymbol{\beta}$ to 0.075.

8. Qualitative Results

As shown in Figure 6, we show some qualitative results of our effective and lightweight network ELPN-V1 on the COCO val2017 dataset [15]. We first used the lightweight human body detector YOLO-V3 [44] to detect the bounding box of the human body, and then used our ELPN-V1 model to detect the specific keypoint locations. We selected some samples with complex actions to display. The results show that our model can detect those keypoints very well, thus verifying that our model is effective.

Figure 6. Illustrations the qualitative pose estimation results on the COCO val2017 dataset, including multi-person and single-person effects. We first used the YOLO-V3 [44] model to detect the human bounding box, and then used our human pose estimation model ELPN-V1 to detect human keypoints. It can be seen that our model can detect the keypoint positions well even in the presence of complex actions.

The first two columns in the figure are the results of multiple people, and the last column is the result of a single person. From the two results in the first column, it can be seen that our model can not only detect the key points with complex actions, but also the key points with self-occlusion. For the field of human pose estimation, it is very challenging to accurately detect the key points of occlusion. Our model can do this to a certain extent. The images in the second column also contain complex actions, and our model can also detect the key points of all of them well. The two images in the third column are both action images of athletes playing badminton, including very complicated actions. Our model has also successfully positioned the key points of its human body. In general, our lightweight model is effective and efficient. It can accurately locate the key points of the human body with the advantages of low parameter amount and low computational overhead. This lays the foundation for using the human body pose estimation model on edge devices, and makes the human body pose estimation have a broader application space.

9. Conclusion

In this paper, we use the top-down pipeline to deal with the multi-person pose estimation problem. First of all, we propose a novel lightweight network structure called Effective and Lightweight Pose Network (ELPN). Secondly, we propose a Multi-Angle Pose Distillation (MAPD) model training method that can more effectively train particularly small pose network models. We directly use the trained large model as the teacher to guide the lightweight student network model training. During the test, the student network is used for testing, and the teacher model is directly discarded. This training method has many advantages, such as fast convergence speed of model training, more stable training process and so on. Compared with most existing methods, our methods achieve more efficient human pose estimation models while ensuring the generalization ability of the model. In quantitative experiments, our models have excellent performance on the MPII and COCO datasets. In qualitative experiments, our models can accurately locate the keypoints of complex human movements. These fully demonstrates the efficiency and effectiveness of our methods. Our models have the characteristics of high precision, small size and fast inference speed. It has greater practical value. We

hope that our work can provide a certain reference value for designing simple, effective and lightweight human pose estimation models.

References

[1] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, X. Wang, 2019. Attention guided unified network for panoptic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **520**, 7026–7035
[2] P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang, G. Huang, 2019. State-aware reidentification feature for multi-target multi-camera tracking, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*
[3] X. Cai, W. Zhou, L. Wu, J. Luo, H. Li, 2015. Effective active skeleton representation for low latency human action recognition, *IEEE Transactions on Multimedia* **18**(2) 141–154.
[4] Z. Fan, X. Zhao, T. Lin, H. Su, 2018. Attention-based multi-view re-observation fusion network for skeletal action recognition, *IEEE Transactions on Multimedia* **21**(2) 363–374.
[5] A. Marcos-Ramiro, D. Pizarro, M. Marron-Romera, D. Gatica-Perez, 2015. Let your body speak: Communicative cue extraction on natural interaction using rgbd data, *IEEE Transactions on Multimedia* **17**(10) 1721– 1732.
[6] K. Sun, B. Xiao, D. Liu, J. Wang, 2019. Deep high-resolution representation learning for human pose estimation, in: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703.
[7] F. Zhang, X. Zhu, M. Ye, 2019. Fast human pose estimation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3517–3526.
[8] A. Bulat, G. Tzimiropoulos, 2017. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources, in: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3706–3714.
[9] Z. Zhang, J. Tang, G. Wu, Simple and lightweight human pose estimation, *arXiv preprint* arXiv:1911.10346.
[10] J. Lu, J. Xuan, G. Zhang, X. Luo, 2018. Structural property-aware multilayer network embedding for latent factor analysis, *Pattern Recognition* **76** 228–241.
[11] J. Lu, H. Zuo, G. Zhang, 2019. Fuzzy multiple-source transfer learning, *IEEE Transactions on Fuzzy Systems* **28**(12) 3418–3431.
[12] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, 2018. Cascaded pyramid network for multi-person pose estimation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112.
[13] W. Yang, S. Li, W. Ouyang, H. Li, X. Wang, 2017. Learning feature pyramids for human pose estimation, in: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1281–1290.
[14] A. Newell, K. Yang, J. Deng, 2016. Stacked hourglass networks for human pose estimation, in: *European Conference on Computer Vision*, Springer, pp. 483–499.
[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, C. L. Zitnick, 2014. Microsoft coco: Common objects in context, in: *European Conference on Computer Vision*, Springer, pp. 740–755.
[16] B. Xiao, H. Wu, Y. Wei, 2018. Simple baselines for human pose estimation and tracking, in: *Proceedings of The European Conference on Computer Vision (ECCV)*, pp. 466–481.
[17] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2014. 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.
[18] X. Chen, A. L. Yuille, 2014. Articulated pose estimation by a graphical model with image dependent pairwise relations, in: *Advances in Neural Information Processing Systems*, pp. 1736–1744.

[19] M. Andriluka, S. Roth, B. Schiele, 2009. Pictorial structures revisited: People detection and articulated pose estimation, in: *2009 IEEE Conference On Computer Vision And Pattern Recognition*, IEEE, pp. 1014–1021.
[20] M. A. Fischler, R. A. Elschlager, 1973. The representation and matching of pictorial structures, *IEEE Transactions on Computer*s **100**(1) 67–92.
[21] K. He, G. Gkioxari, P. Doll´ar, R. Girshick, Mask r-cnn, 2017. in: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969.
[22] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, 2017. Realtime multi-person 2d pose estimation using part affinity fields, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299.
[23] A. Newell, Z. Huang, J. Deng, 2017. Associative embedding: End-to-end learning for joint detection and grouping, in: *Advances in Neural Information Processing Systems*, pp. 2277–2287.
[24] M. Kocabas, S. Karagoz, E. Akbas, 2018. Multiposenet: Fast multi-person pose estimation using pose residual network, in: *Proceedings of The European Conference on Computer Vision (ECCV)*, pp. 417–433.
[25] A. Toshev, C. Szegedy, 2014. Deeppose: Human pose estimation via deep neural networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660.
[26] J. Carreira, P. Agrawal, K. Fragkiadaki, J. Malik, 2016. Human pose estimation with iterative error feedback, in: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4733–4742.
[27] K. He, X. Zhang, S. Ren, J. Sun, 2016. Deep residual learning for image recognition, in: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
[28] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, X. Wang, 2017. Multicontext attention for human pose estimation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1831–1840.
[29] K. Su, D. Yu, Z. Xu, X. Geng, C. Wang, 2019. Multi-person pose estimation with enhanced channel-wise and spatial information, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5674–5682.
[30] J. Hu, L. Shen, G. Sun, 2018. Squeeze-and-excitation networks, in: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141.
[31] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond, in: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0.
[32] X. Wang, R. Girshick, A. Gupta, K. He, 2018. Non-local neural networks, in: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803.
[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv preprint* arXiv:1704.04861.
[34] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint* arXiv:1412.6980.
[35] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, 2017. Towards accurate multi-person pose estimation in the wild, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4903–4911.
[36] X. Sun, B. Xiao, F. Wei, S. Liang, Y. Wei, 2018. Integral human pose regression, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 529–545.
[37] S. Yang, W. Yang, Z. Cui, Pose neural fabrics search, *arXiv preprint* arXiv:1909.07068.
[38] J. J. Tompson, A. Jain, Y. LeCun, C. Bregler, 2014. Joint training of a convolutional network and a graphical model for human pose estimation, in: *Advances in Neural Information Processing Systems*, pp. 1799–1807.

[39] U. Rafi, B. Leibe, J. Gall, I. Kostrikov, 2016. An efficient convolutional network for human pose estimation., in: *BMVC*, **1**, 2.
[40] V. Belagiannis, A. Zisserman, 2017. Recurrent human pose estimation, in: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, IEEE, pp. 468–475.
[41] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, B. Schiele, 2016. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, in: European Conference on Computer Vision, Springer, pp. 34–50.
[42] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, 2016. Convolutional pose machines, in: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4724–4732.
[43] W. Tang, P. Yu, Y. Wu, 2018. Deeply learned compositional models for human pose estimation, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 190–206.
[44] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, *arXiv preprint* arXiv:1804h.02767.