Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **5**, Number **4**, November **2020**

Lightweight Human Pose Estimation Network and Angle-based Action Recognition

Guang-Yu Kang

School of Control and Mechanical Engineering
Tianjin Chengjian University
Tianjin 300222, P.R. China
kgycat250318@sina.com

Zi-Qian Lu and Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
*Correponding Author: zheminglu@zju.edu.cn

Received March 2020;Revised October 2020

Abstract. *Human pose estimation and action recognition is challenging. There are many end-to-end convolutional neural network-based methods for pose estimation, but they are too difficult and costly to train. We propose an improved lightweight network for pose estimation, which greatly reduces the difficulty and time of training. It can also greatly improve Frames Per Second(FPS). In the meanwhile, based on the pose estimation, the cosine theorem is used to analyze the angle information of key points in real time. Finally, with comprehensive information, our method achieves rapid action recognition and does not require training in advance. We evaluate our models on standard pose estimation benchmarks and show that proposed lightweight models outperform base deterministic ones.*
Keywords: Pose estimation, Lightweight network, Cosine theorem, Angle information, Action recognition.

1. **Introduction.** Pose estimation and action recognition is a hot research area in computer vision. This problem is a key to successful analysis of people's action on video. In the early years, it relied mainly on a camera with a depth sensor (RGB-D camera) to capture the human body posture in real time, but this method has many drawbacks. First of all, due to the distance limitation, the human body pose estimation can only be carried out within a certain distance of the camera, and the out-of-range will be invalid. Secondly, all application scenarios must be equipped with complex depth sensing devices or depth sensing, and the expensive equipment increases the complexity and cost of the tasks. Thirdly, it is difficult to deal with occlusion problem and has low robustness. Therefore, this method is difficult to be used in multiplayer pose estimation.

To solve these problems, many algorithms based on deep learning have emerged in recent years to solve videos and pictures pose estimation and action recognition tasks. The deep learning methods have solved some of the inherent problems faced by some human pose estimation algorithms. For example, the occlusion of clothes, the occlusion between people, the change of illumination, the change of camera angle, etc., and it has achieved good results in terms of accuracy and robustness. It also derived a variety of

different network structures. However, there are some drawbacks to using deep learning methods to solve such problems. For examples, the training parameters are too large, the training time is too long, and the generalization of tasks is not strong, etc.

At present, some pose estimation algorithms based on deep learning can be roughly divided into two categories. One is a top-down estimation method. The operation steps are generally to detect the bounding box of the human body and then convert the problem into a single-person pose estimation. The other is the bottom-up method, which calculates the information such as the position of each key point and then combines them together. It can be seen that the basic idea of any method is to perform feature extraction, letting the network update the weight continuously, and learn to estimate the posture information we need from the video stream. There are many networks for extracting image features, from early AlexNet [1] to VGG [2] to ResNet [3]. These network structures have achieved remarkable success in image feature extraction tasks, but achieving these successes requires deeper and deeper layers of network and exponential growth parameters, such as AlexNet [1] network have 8 layers, a total of 60 million parameters and need to consume about 237.9545MB of memory, VGG16 network have 16 layers, a total of about 138 million parameters and need to consume about 527.7792MB of memory. The network structure parameters of the ResNet [3] series are even more difficult to calculate. Training time also increases with the complexity of the network structure. These problems are unacceptable to many researchers and users. In order to solve these problems, there are two ways. The first one is to use more higher performance graphics cards, but the high cost of this method does not apply to everyone. The second is to optimize the network structure and reduce the space and time complexity with a light weight network. This paper uses the second method.

The main contribution of this article is to propose the use of lightweight network structure for training, which greatly reduce the difficulty of training, significantly improve the Frames Per Second, and realize the task of real-time human pose estimation. After obtaining the key points of the human body, simple action recognition is accomplished through information such as position and angle between the key points. The example is shown in Figure 1.

The structure of this paper is as follows. In Section 2, we describe the recent human body pose estimation work. Then we describe the MobileNet [4] based pose estimation network in Section 3. In Section 4, we give action recognition based on multiple information co-occurrence. In Section 5, we give experimental results and analysis. We make a conclusion to our work in Section 6.

2. **Related Work.**

2.1. **Human pose estimation by lightweight network.**

The methods of the human body pose estimation field before 2015 are all based on the accurate joint point coordinates (x, y). It is easily affected by the flexibility of human movement, and the model is less scalable. From the 2015 Flow Convnet [5], the human pose estimation is regarded as the detection problem, and the output is heatmap. In the FLIC dataset, this method can achieve 92% of the average PCK for wrist and elbow, although only half-body key points. CPM [6] is one of the most popular methods for human pose estimation. Newell et al.[7] introduced the hourglass network, which is the basis of many other jobs. This method is designed to capture different scale features of the input image and combine them with repeated down-sampling and up-sampling. State-of-the-art approaches for human pose estimation are Alphpose [8] Openpose [9] and Mask-rcnn [10]. Alphapose achieved state of the art performance on MPII human pose benchmark and COCO dataset. In the

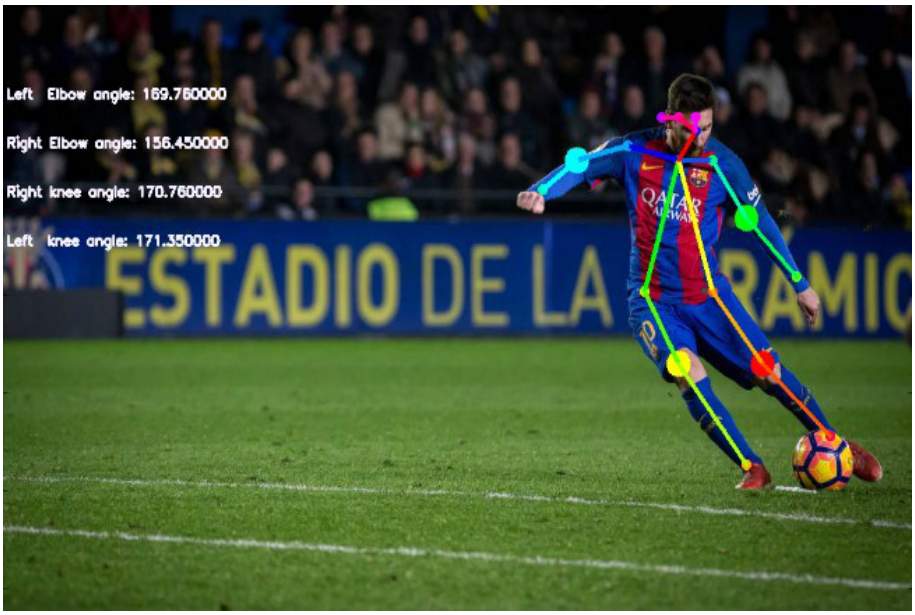


FIGURE 1. The example of pose estimation output with angle information and important key points

meanwhile, there are four main directions for designing lightweight neural network models in industrial and academic circles: (1) Artificially designed lightweight neural network models; (2) Neural network based on Neural Network Architecture Search (NAS);(3) CNN model compression;(4) Automated model compression based on AutoML. Where SqueezeNet [11] , MobileNet [4], ShuffleNet [12], Xception [13] etc. are popular lightweight network structures. SqueezeNet [11] proposes fire modules with squeeze layer and expand layer in a similar way to VGG [2]. The MobileNet [4] proposed by the Google uses the convolution method of depth-wise separable convolution instead of the traditional convolution method to greatly reduce the weight parameters. However, this also brings a problem that the information is not smooth, resulting in the output feature map containing the information of the input feature map. In order to solve this problem, Face++ proposed a network called ShuffleNet [12] to disorderly order the channels of the feature maps of each part to form a new feature map to solve the problem of "Not Smooth Information Flow" brought by group convolution. Xception [13] is based on Inception-v3 [13] to improve network efficiency and achieve higher performance under the same number of parameters.

2.2. **Action recognition.** In the early stage of human action recognition, there are calculation methods based on human body geometry [13] and extraction methods of action information [16], etc. Common datasets are Weizmann [17], UCF [18], etc. Although the algorithm is very slow, state-of-the-art approach for action recognition before using deep learning is iDT [19]. Two Stream Network [20] and derivative methods are relatively mainstream type of deep learning based. The basic principle of Two Stream is to calculate a dense optical stream for every two frames in a video sequence to obtain temporal information. Then the CNN model is trained separately for the video image and the dense optical flow, and the networks of the two branches respectively judge the category of the action. C3D [21], Network also achieved good results.

An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data [22] shifts the focus of work to the processing of skeleton data. In this

paper two LSTM subnetworks in spatio and temporal are designed to give different attention. LSTM also used in Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks [23] and Online Human Action Detection Using Joint Classification-Regression Recurrent Neural Networks [24]. St-gcn [25] proposed a bone-based action recognition network structure, using GCN instead of CNN to achieve good results on NTU Kinects datasets. There are also some other methods to extract features for recognition[26][27][28][29][30][31].

3. **Proposed Modifications for Pose Estimation.**

We apply the lightweight network MobileNet [4] to the human pose estimation framework Openpose [9]. In this way, it greatly reduces the training parameters and speeds up the algorithm.

3.1. **Base algorithms.**

This subsection presents a brief description of two baseline models: Openpose [9] and MobileNet [4], which are used in our methods.

3.1.1. *Openpose.* Openpose [9] is one of the stat-of-art human pose estimation approaches proposed by CMU. The RGB image with the size of $w \times h$ was used as input, and a feature map F was obtained through the first 10 layers network of VGG. In the first stage, the network is divided into two cyclic branches with feature map F as the input and in the remaining stages the output of the previous layer and F are treated as inputs. One branch is used to predict confidence map S (human body joints), and one branch is used to predict L (PAFs). Each branch will calculate the L_2 norm of L and S and get the sum of the loss function f in each layer. The loss functions at both branches at stage t and sum of loss functions are:

$$f_S^t = \sum_{j=1}^{J} \sum_{P} W(P) \cdot \|S_j^t(p) - S_j^*\|_2^2 \tag{1}$$

$$f_L^t = \sum_{c=1}^{C} \sum_{P} W(P) \cdot \|L_c^t(p) - L_c^*\|_2^2 \tag{2}$$

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \tag{3}$$

For the CMP section, through the 2D point $X_{j,k}$ in the image, where $X_{j,k}$ represents the j_{th} joint of the k_{th} person in the picture. Where the groundtruth $S_{j,k}^*$ conforms to the normal distribution. When the pixel point P approaches the annotation point $X_{j,k}$, it reaches to the peak of the normal curve.

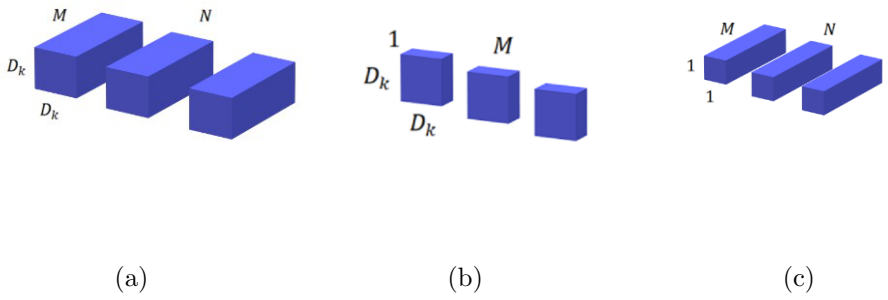
For the PAF section, Calculate the ground truthL_c^* of L by the unit vector of any pixel P between the two points of the k_{th} person, $X_{j1,k}$, $X_{j2,k}$, where k represents the k_{th} person, and $j1$ and $j2$ represent the number of two joints that can be connected (eg elbows and wrists are connected by arms), c indicates c_{th} limbs.

After the CMP and PAF are obtained, the key points are regarded as the vertices in the graph and the PAF is regarded as the weight of the edge. Multi-person detection problem is transformed into bipartite graph matching problem, and the Hungarian algorithm is used to obtain the optimal matching of connected key points.In this method, the parallel calculation of PAF and CMP makes the result significantly improved, so we also adopt a similar structure.

3.1.2. *MobileNet.* MobileNet [4] is a deep decomposable convolution model originally proposed by Google for the mobile-end model. The use of deep separable convolution involves two hyperparameters to reduce the amount of parameters and the amount of computation:(1) width multiplier for reducing input and output channels, (2) resolution multiplier for reducing the feature map size of the input and output. The standard convolution convert the input layer with dimension $D_I \cdot D_I \cdot M$ into dimension $D_O \cdot D_O \cdot N$ shown in Figure 2(a). Assuming that the size of the convolution kernel filter is $D_k \cdot D_k$, the whole calculation of the standard convolution is: $D_I \cdot D_I \cdot M \cdot N \cdot D_k \cdot D_k$. For deep decomposable convolution, depthwise convolution and pointwise convolution are required shown in Figure 2 (b) and (c). The size of the deep convolutional convolution kernel is $1 \cdot M \cdot D_k \cdot D_k$, so the total calculation amount is $D_I \cdot D_I \cdot M \cdot D_k \cdot D_k$, and the size of the point-by-point convolutional convolution kernel is $1 \cdot 1 \cdot N \cdot M$, so the total calculation amount is: $D_I \cdot D_I \cdot M \cdot N$. So the ratio of separable convolution to the complexity of traditional convolution is as follows:

$$\frac{D_I \cdot D_I \cdot M \cdot D_k \cdot D_k + D_I \cdot D_I \cdot M \cdot N}{D_I \cdot D_I \cdot M \cdot N \cdot D_k \cdot D_k} = \frac{1}{N} + \frac{1}{D_k^2} \tag{4}$$

As we can see, this method reduces the computation sharply.

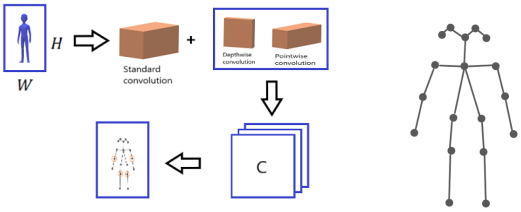


(a) (b) (c)

FIGURE 2. The structure of the convolution layer (a) Standard convolution (b)Depthwise convolution (c)Pointwise convolution

3.2. **Our Methods.** In this subsection, we propose a combination of bottom-up pose estimation and lightweight networking to achieve the goal of quickly completing the estimation task, and after getting the skeleton key points, we add joint angles and position information to prepare for the next action recognition.

3.2.1. *Network structure.* In CMU works, the image is first analyzed by a convolutional network (initialized by the first 10 layers of VGG-19 and finetuned) and get a feature map with 128 dimensions. We use 11 depth separable convolutions and 1 standard convolution layer instead of VGG [2]. Six stages of separable convolution (each stage consists of five convolution layers) are used for further extraction from previously image features. And we optimize the predictive value and true value between the L_2 norm to make the network more accurately. The model structure can be seen in Figure 3. It is worth mentioning that we have received a good result by using a separate convolution instead of a traditional convolution, which we will introduce in more detail below. Finally, we obtain the prediction of the 18 skeleton key points of the human body, and use the coco dataset's paradigm to link them.



(a) (b)

FIGURE 3. The model structure of lightweight network

3.2.2. *Cosine angle information.* When our network calculates the position of the key points of the human skeleton, it usually reveals only a tree-shaped body structure. However, it's not usually very useful to do this, because the spatial syntax information can only provide a primary information, and it's not very helpful in the following tasks of action recognition, behavioral analysis, and so on. Therefore, we propose to use the cosine theorem to calculate the Angle information between partial joints, including the two joint angles of the upper arm and the two joint angles of the lower leg. This will provide more information for action recognition.

After extracting features from the previous part of the lightweight network, we connected 18 key points of the human body into a tree structure. Each of these joints is made up of three adjacent nodes shown in Figure 4,5 and these formulas shown below are used to calculate joint angles.

$$a = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5}$$

$$a = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \tag{6}$$

$$a = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \tag{7}$$

$$\theta = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \tag{8}$$

Where a, b and c represent the lengths of the limbs, and θ represents the joint angle of three adjacent nodes.

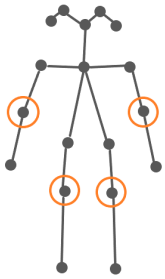


FIGURE 4. The human tree structure

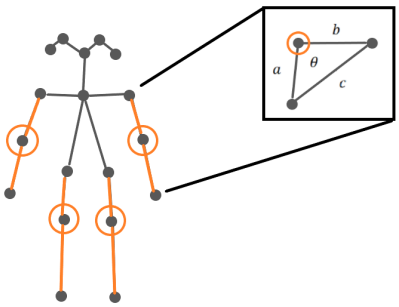


FIGURE 5. The angle of the joints

Then, the output is the position prediction of the key points of the human body and the four joint angles.

4. **Simple Action Recognition.** Using our lightweight network model, we have got 18 key location information and four angles information of joints, which makes our action recognition task much easier. As an example, we determine that the body is in running status when the arms and legs are at different sides and the angle of the knee joint of the support leg changes from an acute angle to an obtuse angle. On the contrary, the human body is in a non-running state. In this way, some action recognition can be implemented quickly on the basis of pose estimation. Compared with the end-to-end action recognition framework, it has great advantages in complexity and speed.

Most importantly, the algorithm that can be implemented without training is very useful for practical applications. By changing the discriminant rules, such as the positional relationship and angular size of the limbs the model can be applied to actions identified by a particular context. There are some examples in Figure 6:

(a) (b)

FIGURE 6. The top two are illustrations of no running, and the middle and bottom are examples of running. The upper left corner of each graph shows the size of their joint angles

5. **Experimental Results.** Tensorflow framework is used for training and RMSProp algorithm is used for optimization. We use NVIDIA 1080Ti GPU for training with batch size of 24, learning rate of 0.01 on COCO dataset. We tested on gtx1050, although the accuracy was not the best, but the speed was greatly improved. The resolution time of the single picture was only about 0.04s, and the frame rate of video was about 20 25fps, while the original method of CMU is only about 3fps. Using MobileNet [4] as the lightweight

network structure mentioned in the article, the model file size after training is only 7.8MB and the model of CMU is 204MB.

COCO is a large-scale object detection, segmentation, and captioning dataset, which has several features: Object segmentation, 330K images (>200K labeled), and 250,000 people with key points. The COCO evaluation defines the object key point similarity (OKS) and uses the mean average precision(AP) over 10 OKS thresholds and average recall(AR) as main competition metric. The OKS plays the same role as the intersection-over-union(IoU) in object detection. It is calculated from scale of the person and the distance between predicted points and ground truth points. Table 1 and Table 2 show results from teams in the challenge16 with COCO2014/Val, where AP^{50} represents average precision with OKS = 50, and AP^M represents average precision with smaller scales, and AP^L represents average precision with larger scales. The superscript of AR has the same meaning as the above AP. Although not the best, it has achieved good results. It is worth noting that our method has a great advantage in speed, reaching about 22fps and basically realizing real-time pose estimation compared with CMU work in Table 3.

Table 1. The AP comparison on the COCO 2016 challenge

Methods	AP	AP^{50}	AP^{75}	AP^M	AP^L
zjhuang	0.105	0.300	0.056	0.082	0.141
Ours	**0.280**	**0.547**	**0.248**	**0.280**	**0.285**

Table 2. The AR comparison on the COCO 2016 challenge

Methods	AR	AR^{50}	AR^{75}	AR^M	AR^L
zjhuang	0.190	0.439	0.142	0.108	0.301
Ours	**0.320**	**0.584**	**0.290**	**0.292**	**0.358**

Table 3. The AR comparison on the COCO 2016 challenge

Methods	FPS	$Parsing$
CMU	5.3	0.304s
Ours	**22.5**	**0.059**

In addition, we also analyzed the reasoning time of the algorithm in the case of multiple people. Figure 7 shows a comparison of our method and CMU work. It can be seen that the reasoning time increases slightly with the increase of the number of people in the way of thinking, and the method of our algorithm is much less time-consuming than the CMU method under the same number of people. Probably the time consumption is only 0.04 times the comparison work. We also compare with the most popular algorithms in the

same situation. It can be seen from Figure 8. that although the CMU method estimates the half-length person on the right side of the picture, compared with our method, we have better predictions for the neck and crotch. In contrast, their methods are predicted to be lower and more right.

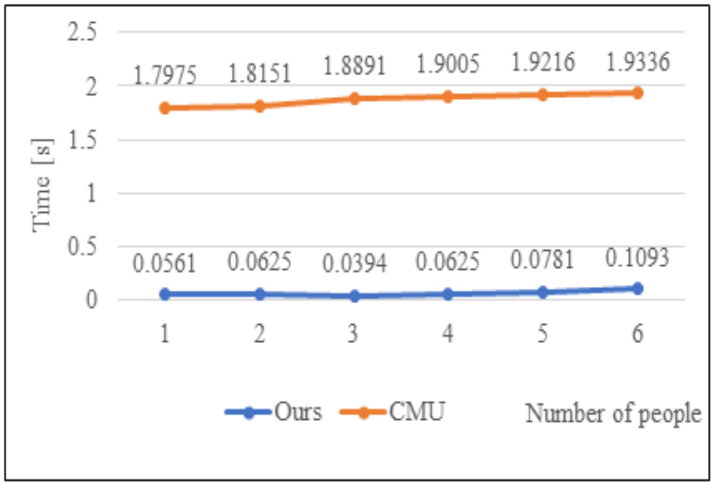
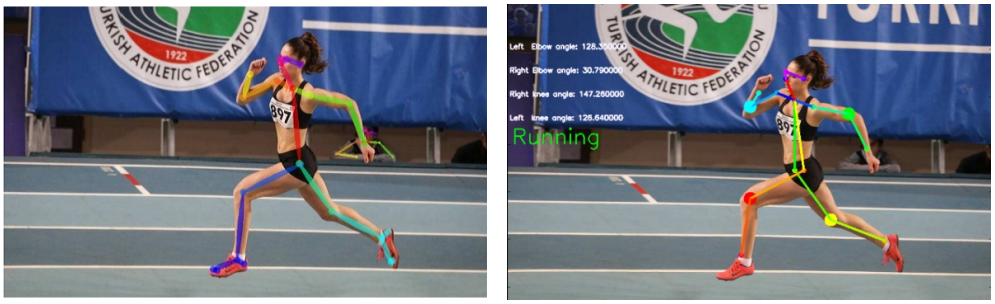


FIGURE 7. Time consumption comparison in multi-person situations



(a) CMU (b) Ours

FIGURE 8. Comparison between CMU and our results

6. **Conclusion.** In this paper, we propose to use the lightweight network as the backbone network to make the pose estimation task more simple and fast. This is necessary in real life applications. In addition, on the basis of pose estimation, we introduced the physical information of key points and made rules to achieve simple action recognition. As is shown in the experiment results, we achieved good results in both tasks compared to the original model.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[2] K. Simonyan, and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.
[4] A. G. Howard, M. Zhu, B. Chen, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv preprint arXiv:1704.04861*, 2017.
[5] T. Pfister, J. Charles, and A. Zisserman, Flowing convnets for human pose estimation in videos, In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1913-1921, 2015.
[6] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, Convolutional pose machines, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724-4732, 2016.
[7] A. Newell, K. Yang, and J. Deng, Stacked hourglass networks for human pose estimation, In *European Conference on Computer Vision*, pp.483-499, Springer, Cham, 2016.
[8] H. S. Fang, S. Xie, Y. W. Tai, and C. Lu, RMPE: Regional Multi-person Pose Estimation, In *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, pp. 2353-2362, 2017.
[9] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291-7299, 2017.
[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, Mask r-cnn, In *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, pp. 2980-2988, 2017.
[11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, Squeezenet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size, *arXiv preprint arXiv:1602.07360*, 2016.
[12] X. Zhang, X. Zhou, M. Lin, and J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
[13] F. Chollet, Xception: Deep learning with depthwise separable convolutions, *arXiv preprint arXiv:1610-02357*, 2017.
[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826, 2016.
[15] H. Fujiyoshi, AJ. Lipton, T. Kanade, Real-time human motion analysis by image skeletonization, *IEICE Transactions on Information and Systems*, vol. 87, no. 1, pp. 113-120, 2004.
[16] R. Chaudhry, A. Ravichandran, G, Hager, et al., Histograms of oriented optical flow and binet-cauchy kernels on nolinea dynamical systems for the recognition of human actions, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1932-1939, 2009.
[17] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, Ronen Basri, Actions as Space-Time Shapes, In *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, pp. 1395-1402, 2005.
[18] K. Soomro, A. R. Zamir, and M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, *arXiv preprint arXiv:1212.0402*, 2012.
[19] H. Wang, and C. Schmid, Action Recognition with Improved Trajectories, In *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2014.
[20] K. Simonyan, and A. Zisserman, Two-stream convolutional networks for action recognition in videos, In *Advances in neural information processing systems* pp. 568-576, 2014.
[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, Learning spatiotemporal features with 3d convolutional networks, In *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, pp. 4489-4497, 2015.
[22] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data, In *AAAI*, vol. 1, no.2, pp. 4263-4270, 2017.
[23] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks, In *AAAI*, vol. 2, no. 5, pp. 6, 2016.
[24] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, Oline human action detection using joint classification-regression recurrent neural networks, In *European Coference on Computer Vision*, pp. 203-220, 2016.
[25] S. Yan, Y. Xiong, and D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, *arXiv preprint arXiv:1801.07455*, 2018.
[26] K. K. Tseng, R. Zhang, C. M. Chen, M. M. Hassan, DNetUnet: a semi-supervised CNN of medical image segmentation for super-computing AI service, *The Journal of Supercomputing*, pp. 1-22, 2020.

[27] E. K. Wang, C. M. Chen, M. M. Hassan, and A. Almogren, A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain, *Future Generation Computer Systems*, vol. 108, pp. 135-144, 2020.

[28] E. K. Wang, C. M. Chen, F. Wang, M. K. Khan, and S Kumari, Joint-learning segmentation in Internet of drones (IoD)-based monitor systems, *Computer Communications*, vol. 152, no. 2, pp. 54-62, 2020.

[29] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, Human motion recognition based on SVM in VR art media interaction environment, *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[30] E. K. Wang, X. Zhang, F. Wang, T.-Y. Wu, C.-M. Chen, Multilayer dense attention model for image caption, *IEEE Access*, vol. 7, pp. 66358-66268, 2019.

[31] F. Zhang, T.-Y. Wu, and G. Zheng, Video salient region detection model based on wavelet transform and feature comparison, *EURASIP Journal on Image and Video Processing*, vol. 2019, no.1, 58, 2019.