Lightweight dynamic large convolution model for real-time human pose estimation

Zongyou Liu*, Shilan Liu, Ziyuan Liu, Hongtao Wang, Quanxin Jin
School of Computer Science, Shenyang Aerospace University, Shenyang, Liaoning, China

ABSTRACT

Although the lightweight Openpose model can perform real-time human pose estimation on the CPU, it also has the following problems: (1) The ReLU activation function destroys the narrow channel information; (2) Although the model has high inference efficiency, it also has weak feature extraction ability; (3) The model has a small effective receptive field. In order to improve the above problems, we adjusted the structure of the lightweight Openpose model, replacing the backbone of the lightweight Openpose with the backbone of MobileNetV2 to improve the destruction of the narrow channel information by the ReLU activation function. Replacing part of the backbone's convolutional layers with dynamic depthwise separable large convolutional layers expands the effective receptive field of the model, improves feature extraction and slightly increases the computational cost. Our Lightweight dynamic large convolution model achieves an Average Precision (AP) value of 0.40 on the COCO2017 validation set.

Keywords: Real-time human pose estimation, computer vision, feature extraction capabilities

1. INTRODUCTION

In the past few decades, deep convolutional neural networks have made great breakthroughs in the field of computer vision [1-3]. One way to construct a deep convolutional neural network is to stack convolutional layers as much as possible. Although deep neural networks have achieved advantages in computer vision, their high computational cost makes it difficult for people to apply them to real-time software. In order to enable neural network models to be applied to real-time software, a variety of lightweight models have been studied to complete real-time tasks such as ShuffleNet [4], MobileNetV1 [5], MobileNetV2 [6], MobileNetV3 [7] and lightweight OpenPose [8]. Although these lightweight models have lower computational cost, there are certain drawbacks: (1) The feature extraction ability is weak. (2) The effective receptive field is small.

In order to improve the above problems, we adjusted the structure of the lightweight OpenPose model and replaced the backbone of the lightweight Openpose with the backbone of MobileNetV2 to reduce the destruction of the narrow channel information by the ReLU activation function. Conditional parameterized large convolutions are used to replace part of the backbone's convolutional layers to expand the effective receptive field [9] and feature processing capabilities of the model.

Our Lightweight dynamic large convolution model has an AP value of 0.40 on the COCO2017 validation set.

2. RELATED WORK

Lightweight model. In recent years, research on lightweight models is a hot field and there are many lightweight models today. MobileNetV1 proposes a depthwise separable convolution to replace the convolutional layer and the depthwise separable convolutional layer greatly reduces the computational cost of the model. MobileNetV2 proposes Inverted residual block, which not only reduces the inference efficiency of the model but also reduces the damage of the ReLU [10] activation function to narrow channels. Lightweight Openpose replaces part of the structure of the OpenPose model with depthwise separable convolution and 3*3 convolution layers to improve inference efficiency. Our model backbone is MobileNetV2.

Large convolution kernel. Since the VGG model was proposed, large convolution kernel models such as inception are no longer popular. Recently ConvNeXt uses 7×7 convolution kernels to create network models. ConvMixer uses a 9×9 convolution kernel to build the network. Ding et al. found that large convolution kernels can improve the effective

* 1820634098@qq.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 1250624
 · 0277-786X · doi: 10.1117/12.2662876

receptive field of the model. Our model refers to the design ideas of Ding et al. to expand the effective receptive field of the model.

Multi-branch convolutional networks. There are currently some network structures such as Inception and ResNext that have demonstrated success in computer vision. Their convolutional layers contain multiple convolutional branches, which are aggregated to compute the final output. Our Conditionally Parameterized Convolutions (CondConv) [11] are mathematically equivalent to multi-branch structures, but they have less computational cost than multi-branch.

3. METHOD

In this part, we will introduce the network structure of the Lightweight dynamic large convolution model, which consists of four parts, namely backbone, Convolution Pose Machines (CPMs), initial stage and refinement stage. Figures 1-4 show the structure of these four parts in turn.

Figure 1. Structure of backbone.

3.1 Backbone structure

Backbone contains convolutional layers, Inverted Residual, CondConv Inverted Residual. The convolutional layer is followed by a BN layer and a ReLU6 activation function. Inverted Residual consists of 3 convolutional layers. Inverted Residual consists of a convolutional layer with kernel size of 1, depthwise separable convolutional layer and a convolutional layer without activation function. The feature map is processed by the first convolutional layer of Inverted Residual to double the number of channels of the feature map and then processed by the depthwise separable convolutional layer. Finally, the convolutional layer without activation function is used to compress the channels. CondConv Inverted Residual can be obtained by replacing all convolutional layers of the Inverted Residual species with conditional parameterized convolutions and performing the corresponding operations. The depthwise separable CondConv of large convolution kernels enlarges the effective receptive field of the model [8].

The mathematical formula of CondConv is as follows:

$$Output(x) = \sigma((\alpha_1 \bullet W_1 + \ldots + \alpha_n \bullet W_n) * x) \quad (1)$$

n represents the number of parallel convolution kernels, α is computed by the learnable routing function and x is the input feature. CondConv first generates α according to the input feature map and multiplies the corresponding convolution kernel by α, and finally aggregates the parallel convolution kernels to form a new convolution kernel to perform the convolution operation on the input feature map. CondConv is equivalent to the multi-branch structure but has less computational cost than the multi-branch structure.

The generation steps of α_i are: global average pooling, fully connected layer, and sigmoid activation function. R represents a learnable matrix that maps pooled inputs to n expert weights.

$$\alpha_i = Sigmoid(GlobalAveragePool(x)\ R) \quad (2)$$

3.2 CPM and initial stage

The CPM and initial stages process the features of the backbone to generate heatmaps and Part Affinity Fields (PAFs). heatmaps represent the confidence of the joints in the image and PAF represents the confidence of the connection of different joints. The CPM part consists of 5 convolutional layers, the first and last convolutional layers are followed by the BN layer and the ReLU activation function and the rest of the convolutional layers are only followed by the ReLU activation function. Figure 2 shows the structure of CPM and the process of processing information.

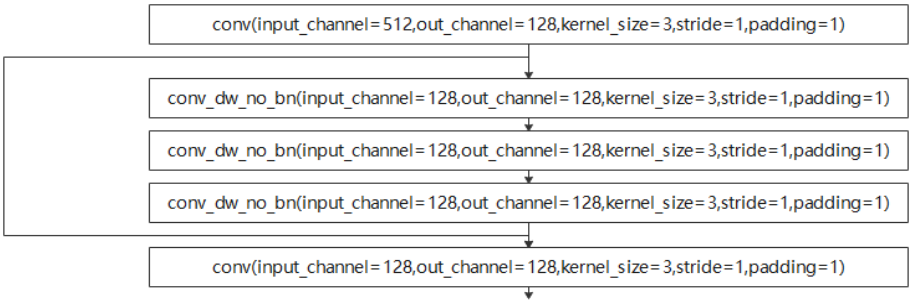

Figure 2. CPM structure.

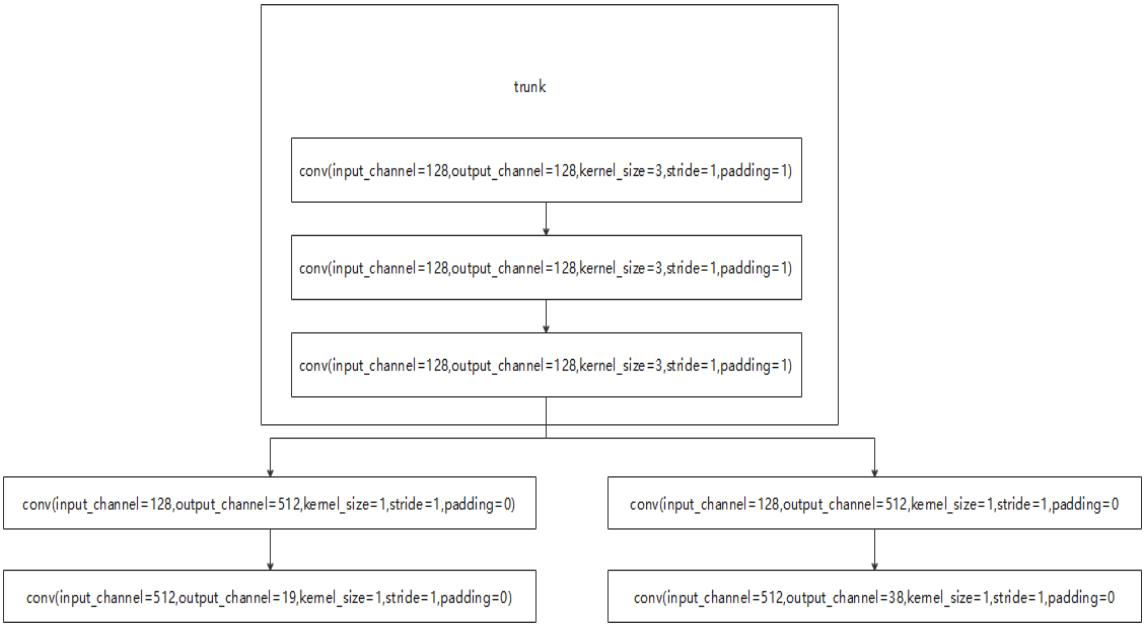

Figure 3. Initial stage structure.

The initial stage contains a trunk and two branches. trunk processes the output of CPM. The trunk contains 3 convolutional layers followed by a BN layer and a ReLU activation function. The two branches deal with the trunk's generated features independently to generate heatmaps and pafs. The two branches have the same structure. The first convolutional layer is followed by only the ReLU activation function and the second convolutional layer directly outputs the result. Figure 3 shows the structure and processing of the initial stage model.

3.3 Refinement stage

The refinement stage integrates the output of the CPM and initial stages to generate more accurate heatmaps and PAFs. The refinement stage consists of trunk and two branches. The trunk contains 5 refinement stage blocks, which contain 3 convolutional layers. The first convolutional layer is followed by only the ReLU activation function and the rest of the convolutional layers are followed by the BN layer and the ReLU activation function. Figure 5 shows the structure and processing of the refinement stage block. The two branches have the same branch structure as the initial stage.

Figure 4. Refinement stage structure.

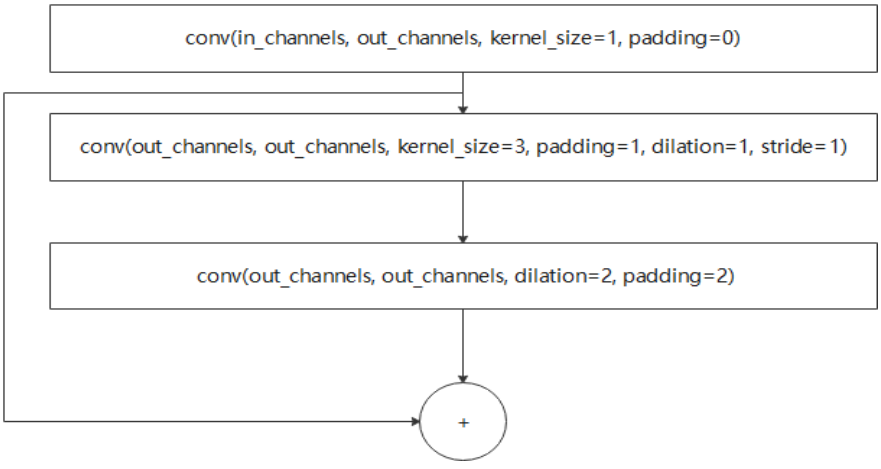

Figure 5. The process of refinement stage blocks processing information.

4. EXPERIMENT

In this part, we use the COCO2017 data set as the basis of the experiment, use the COCO2017 training set to train the model and observe the AP value of the trained model on the COCO2017 validation set. In order to study the influence of the number of first-layer convolutional output channels of Inverted Residual on the accuracy of the model, we conducted comparative experiments.

Training details. We set the training batch size to 32 and stop training at epoch 260. We train the model with the Adam optimizer with a learning rate of 4e-5 for all parameters of the model.

Comparative experiment. We set the number of output channels of the model's Inverted Residual first-layer convolution to 1 × the number of input channels, 2 × the number of input channels, and 3 × the number of input channels. Table 1 shows the AP values of models with different parameters on the COCO2017 validation set.

Table 1. AP values of different models on the COCO2017 validation set.

Model name	Ap
Lightweight dynamic large convolution model (1 ×)	0.38
Lightweight dynamic large convolution model (2×)	0.40
Lightweight dynamic large convolution model (3×)	0.42

Experimental results. From Table 1, we learned that the larger the number of Inverted Residual first layer convolution output channels, the higher the accuracy of the model. When the number of Inverted Residual first layer convolution output channels is 2 × the number of input channels, the AP value of the model is 0.4. In order not to increase the computational cost and occupy the memory too much, we set the number of output channels of the first layer convolution of all Inverted Residual to 2 × the number of input channels.

5. CONCLUSION

This paper proposes a Lightweight dynamic large convolution model to complete the task of real-time human pose estimation. Our model improves the following problems. (1) The destruction of the narrow channel information by the ReLU activation function is alleviated. (2) The effective receptive field of the model is enlarged. (3) The complexity of the model is increased without increasing the computational cost too much.

REFERENCE

[1] Krizhevsky, A., Sutskever, I. and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25, (2012).

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

[3] He, K., Zhang, X., Ren, S., et al., "Deep residual learning for image recognition," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 770-778 (2016).

[4] Zhang, X., Zhou, X., Lin, M., et al., "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," arXiv preprint arXiv:1707.01083, (2017).

[5] Howard, A. G., Zhu, M., Chen, B., et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, (2017).

[6] Sandler, M., Howard, A., Zhu, M., et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 4510-4520 (2018).

[7] Howard, A., Sandler, M., Chu, G., et al., "Searching for mobilenetv3," Proc. of the IEEE/CVF Inter. Conf. on Computer Vision, 1314-1324 (2019).

[8] Cao, Z., Simon, T., Wei, S. E., et al., "Realtime multi-person 2d pose estimation using part affinity fields," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 7291-7299 (2017).

[9] Ding, X., Zhang, X., Zhou, Y., et al., "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNS," arXiv preprint arXiv:2203.06717, (2022).

[10] Glorot, X., Bordes, A. and Bengio, Y., "Deep sparse rectifier neural networks," Journal of Machine Learning Research 15, 315-323 (2011).

[11] Yang, B., Bender, G., Le, Q. V., et al., "CondConv: Conditionally parameterized convolutions for efficient inference," arXiv preprint arXiv:1904.04971, (2019).