Target recognition algorithm based on improved depth separable convolution

Jun Shi a,*, Yunqing Liu b, Quanyang Liuc, Qiong Zhangd, Yan Feie
Changchun University of Science and Technology Changchun, China
*,a1607332877@qq.com, b mzlyq@cust.edu.cn, c2421966742@qq.com,
d1119365209@qq.com, eyanf@cust.edu.cn

Abstract—Realizing accurate target recognition in the fields of drones, vehicle-mounted systems, robots, etc. has increasingly become a hot research direction for researchers. Aiming at the problem of using deep learning algorithms to identify targets on mobile devices with limited size, the network model has a large number of parameters resulting in a large calculation cost, a complicated recognition process, and a large number of iterations resulting in a long recognition time. This paper proposes an improved deep separable convolution algorithm, which consists of deep convolution and point-by-point convolution, which is based on convolution kernels for step-by-step filtering and combination. After deep convolution, batchnorm is added, and the activation function uses a linear function. Add batchnorm and nonlinear activation function ReLU after 1×1 convolution to ensure the accuracy of target detection. This algorithm reduces the amount of parameters while ensuring recognition accuracy, reduces the amount of recognition calculations, and reduces the weight of the network model. It provides a network foundation for implementing image recognition on devices with limited size.

1. Introduction

Target recognition has always been a hot research issue in computer vision and machine learning. In 2012, Convolutional Neural Network (CNN) achieved great success in the classification task of ImageNet [1], and the target detection method based on Convolutional Neural Network was gradually developed by the majority of researchers and engineers. As the recognition scene becomes more and more complex, the trend of recognition tasks is developing toward deeper and larger convolutional neural networks [3,4,5,6]. Object detection methods based on deep learning mainly use convolutional neural networks for feature extraction, which has a good realization effect in terms of accuracy and multi-target detection and recognition. However, with the gradual development of neural networks, the number of network layers continues to increase. The complex feature information extraction network and the large number of parameters will have problems such as high calculation cost and too many iterations. Therefore, it requires a long time of training and high-configuration hardware support, which makes some application scenarios exist. Limitations, can only be used in certain occasions, can not meet the hardware requirements of these networks on embedded and mobile devices.

For these reasons, lightweight networks have been sought after by researchers and engineers in recent years. Lightweight network implementation methods include compression pre-training network and direct training small network. For compression pre-training model, there are four different methods: 1) pruning [7, 8, 9], the purpose is to remove unimportant parameters and turn the weight matrix into a sparse matrix. 2) Tensor decomposition [10,11], using the channel or spatial redundancy of the weight...
matrix to find their low rank approximation. 3) Quantization [12, 13], using low bits instead of floating point for each weight parameter. 4) Knowledge distillation [14,15], transfer the knowledge in the teacher model to the lightweight student model. Although these methods can effectively compress the neural network into a lightweight network. However, the performance of these methods relies heavily on a given pre-trained model. If there is no improvement in architecture, its accuracy will not be improved further.

Therefore, in order to make the recognition task can be executed on a lightweight hardware platform and ensure the accuracy of recognition, this paper proposes an improved deep separable convolution algorithm, which greatly reduces the amount of parameters compared to ordinary convolution, making the calculation amount With this reduction, the weight of the model is ensured, so that the recognition task can be performed on a specific hardware platform.

2. Network Framework

2.1. Deep Separable Convolution

For the traditional convolution calculation, all channels in the corresponding image area need to be considered at the same time when performing the convolution operation. Depth separable convolution proposes a way to deal with the corresponding area of the image and the channel separately, first consider the regional features, and then consider the channel features. Depth separable convolution is to decompose the standard convolution integral into a depthwise convolution and a 1*1 convolution. 1*1 convolution is also called pointwise convolution. The standard convolution calculation process is shown in Figure 1(a), and its block diagram is shown in Figure 2(a). $D_F$ is the width and height of the input feature map. The input feature map size is:

$$D_F \times D_F \times M$$  \hspace{1cm} (1)

$D_G$ is the width and height of the input feature map. The size of the output feature map is:

$$D_G \times D_G \times N$$  \hspace{1cm} (2)

$D_F = D_G$, $D_K$ is the spatial dimension of the rectangular convolution kernel.

The size of the filter is $D_K \times D_K$, $M$ and $N$ are the number of input and output channels, respectively. The standard convolution kernel size is:

$$D_K \times D_K \times M \times N$$  \hspace{1cm} (3)

The calculation formula of the output characteristic graph $G$ is:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \ast F_{k+i-1,l+j-1,m}$$  \hspace{1cm} (4)

Therefore, the calculation of standard convolution is:

$$D_K \times D_K \times M \times N \ast D_F \ast D_F$$  \hspace{1cm} (5)

For depth separable convolution, first, the depth convolution process is shown in Figure 1(b). The size of its convolution kernel is:

$$D_K \times D_K \times M$$  \hspace{1cm} (6)

The calculation formula of the output feature map is:

$$G_{k,l,m} = \sum_{i,j} K_{i,j,m} \ast F_{k+i-1,l+j-1,m}$$  \hspace{1cm} (7)

In equation (7), $G$ is the output feature map; $K$ is the convolution kernel; $F$ is the input feature map; $i$, $j$ are the feature map pixel positions; $k$, $l$ are the output feature map resolution; $m$ is the number of channels. The calculation of depth convolution is:

$$D_K \times D_K \times M \ast D_F \ast D_F$$  \hspace{1cm} (8)

The point-by-point convolution process is shown in Figure 1(c). The size of its convolution kernel is:

$$1 \ast 1 \ast M \ast N = M \ast N$$  \hspace{1cm} (9)

The calculation amount of point-by-point convolution is:

$$M \ast N \ast D_F \ast D_F$$  \hspace{1cm} (10)
Therefore, the total calculation of the depth separable convolution is:

\[ D_K * D_K * M * D_F * D_F + M * N * D_F * D_F \]  \hspace{1cm} (11)

The ratio of the calculation of the depth separable convolution and the standard convolution is:

\[ \frac{D_K * D_K * M + D_F * M + N * D_F * D_F}{D_K * D_K * M * N + D_F * D_F} = \frac{1}{N} + \frac{1}{D_K} \]  \hspace{1cm} (12)

Since the depth convolution uses a 3 × 3 filter, the amount of calculation can be reduced to 8-9 times \((1/N + 1/9)[16]\).

---

Fig. 1. Convolution process diagram of standard convolution, depth convolution and pointwise convolution

2.2. Improved Depth Separable Convolution

In the ordinary depth separable convolution, the depth separable convolution uses the ReLU activation function after the depth convolution and point-by-point convolution. The ReLU activation function alleviates the gradient problem to a certain extent, which can prevent partial overfitting and accelerate convergence. At the same time, the BN layer is added before the ReLU activation function, which can prevent gradient explosion, accelerate the convergence speed of the network and improve the network accuracy, as shown in Figure 2(b).
The calculation formula of ReLU activation function is:

\[ f(x) = \begin{cases} 
0 & , x < 0 \\ 
x & , x \geq 0 
\end{cases} \]  \hspace{1cm} (13)

The curve of the ReLU activation function is shown in Figure 3.

As can be seen from Figure 3, the ReLU activation function is a piecewise linear function, and its output is 0 for all negative values and 0 of the input, and all positive values remain unchanged, which can only play a unilateral suppression role. The output results are all 0, resulting in the loss of a lot of feature information. Therefore, we remove the ReLU activation function after depth convolution and use linear output.) As shown. The linear output is expressed as follows:

\[ f(x) = Ax + b \]  \hspace{1cm} (14)
Among them, A represents weight, b represents offset, x represents input, and f(x) represents output. Therefore, we add batchnorm and linear activation function after the deep convolution layer, and add batchnorm and ReLU activation function after 1×1 convolution, but the final fully connected layer does not have a nonlinear activation function, but is directly input to the softmax layer classification.

3. Experiment analysis

3.1. Experimental Environment
The experimental software environment is Windows 10 operating system, and the deep learning framework is tensorflow. The experimental hardware environment is an Intel Core i5 quad-core processor, 16 GB of memory, and the GPU is NVIDIA(R) GeForce GTX 1660.

3.2. Experimental Results and Analysis
The experiment selects the pre-trained model on the ImageNet[17] dataset to initialize the model parameters, which can accelerate the model convergence and obtain a better recognition effect. GoogleNet [18] and VGG16 [19] are compared with the improved deep separable convolutional network in this paper. The method in this paper is comparable to VGG16 in accuracy, but it is 34 times smaller than VGG16 in size and 26 times lower in calculation intensity than VGG16. It is 1.5% higher than GoogleNet in accuracy, 1.7 times smaller in size than GoogleNet, and 2.7 times smaller in calculation. The results are shown in Table 1.

| Model       | Accuracy | Mult-Adds | Parameter |
|-------------|----------|-----------|-----------|
| GoogleNet   | 69.7%    | 1550      | 6.9       |
| VGG16       | 71.4%    | 15300     | 140       |
| Article Method | 71.2%    | 582       | 4.1       |

As shown in Table 2, the experiment uses two basic network frameworks, SSD and Faster R-CNN, combined with three feature extraction networks to conduct the ablation experiment of the target detection algorithm, compare the model performance, and verify the effectiveness of the improved algorithm in this paper.

| Model       | VGG16 | Depthwise Separable Conv | Article Method | Accuracy | Parameter |
|-------------|-------|--------------------------|----------------|----------|-----------|
| SSD         | √     | √                        |                | 94.54%   | 497.3     |
| SSD         | √     | √                        |                | 93.69%   | 28.1      |
| SSD         | √     | √                        |                | 94.02%   | 30.2      |
| Faster R-CNN | √     | √                        |                | 96.46%   | 145.7     |
| Faster R-CNN | √     | √                        |                | 95.28%   | 8.5       |
| Faster R-CNN | √     | √                        |                | 95.66%   | 8.6       |

It can be seen from the analysis in Table 2 that the results of combining the basic framework SSD and Faster R-CNN with the feature extraction network can be seen that the improved depth separable convolution algorithm proposed in this paper and the two frameworks have a similar recognition than the VGG16 algorithm. The accuracy, but the calculation amount is obviously reduced a lot, and compared with the ordinary depth separable convolution, the algorithm in this paper is similar to the calculation amount, but the accuracy is improved.
4. Conclusion
The improved deep separable convolution algorithm proposed in this paper can realize the lightening and miniaturization of the recognition network when performing image recognition tasks. Compared with the ordinary convolution, the convolution in this paper uses deep convolution and $1 \times 1$ The combination of convolution instead of $3 \times 3$ convolution greatly reduces the amount of parameters. Solve the problem that the output of deep convolution may cause the loss of information by using a nonlinear activation function, and improve the feature extraction ability of the network. The algorithm in this paper makes the network lightweight while ensuring the accuracy of recognition, and provides an effective method for image recognition scenarios with specific needs such as drones, vehicle-mounted systems, and robots.

REFERENCES
[1] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with DeepConvolutional Neural Networks [J]. Advances in Neural Information ProcessingSystems, 25(2):2012.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015.
[4] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Con-ference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
[6] A. Aghasi, N. Nguyen, and J. K. Romberg. Net-trim: Convex pruning of deep neural networks with performance guaran-tee. In NIPS, 2017.
[7] W. Chen, Y. Zhang, D. Xie, and S. Pu. A layer decomposition-recomposition framework for neuron pruning towards accurate lightweight networks. In AAAI, 2019.
[8] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerat-ing very deep neural networks. In ICCV, 2017.
[9] Y. Kim, E. Park, S. Y oo, T. Choi, L. Yang, and D. Shin. Com-pression of deep convolutional neural networks for fast and low power mobile applications. In ICLR, 2016.
[10] W. Wen, C. Xu, C. Wu. Wang, Y. Chen, and H. Li. Co-ordinating filters for faster deep neural networks. In ICCV, 2017.
[11] I. Hubara, M. Courbariaux, D. Soudry, R. Elyaniv, and Y. Bengio. Quantized neural networks: Training neural net-works with low precision weights and activations. Journal of Machine Learning Research, 18(187):1–30, 2016.
[12] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neu-ral networks. In ECCV, 2016.
[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In NIPS, workshop, 2014.
[14] B. B. Sau and V. N. Balasubramanian. Deep model com-pression: Distilling knowledge from noisy teachers. CoRR,abs/1610.09650, 2016.
[15] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmity Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreaetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural net-works for mobile vision applications. CoRR, abs/1704.04861, 2017.
[16] Zhang J, Jin X, Sun J, et al. Spatial and semantic convolitional features for robust visual object tracking. Multimedia Tools and Applications, 2018.
[17] Olga Russakovsky, Jia Deng, Hao Su, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.
[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

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