A Compact Depth Separable Convolutional Image Filter for Clinical Color Perception Test

Zheyi Wen[1], Chenlu Ye[2], Ming Zhao[1], Fang-Chuan Ou Yang[3]*

[1] School of Computer Science, Yangtze University, China
[2] School of Economics and Management, Yangtze University, China
[3] Department of Digital Multimedia Design, National Taipei University of Business, Taiwan
Wzy422886142@hotmail.com, 2022710010@yangtzeu.edu.cn, hitmzhao@gmail.com, ouyang@ntub.edu.tw

Abstract

Deep convolutional neural networks have achieved good performance in the application of computer vision, but there are also problems, such as a large amount of computation, time consuming, and high memory demand. In this paper, a depthwise separable convolution filter pruning method based on PCA is proposed. First, this paper uses depthwise separable convolution to replace the conventional convolution in ResNet to reduce the number of parameters and the amount of computation in the network. The specific operation process is to first use depthwise convolution to separate the spatial dimension to increase the network width and expand the range of feature extraction, and then use pointwise convolution to reduce the computational complexity of conventional convolution operation. Second, PCA is used to distinguish stacked similar filters and perform dimensionality reduction, which not only alleviates the dimensional disaster, but also achieves compression of data and minimizes information loss. Experimental results show that this method can significantly improve the calculation speed and accuracy of the deep convolutional neural network model, and further compress the model size. On the clinical Color Perception Test Chart, this method reduced the amount of model parameters and MACs on ResNet by about 91% while maintaining the test accuracy at about 95%. With almost no loss of accuracy, this method greatly shortened the running time of the model.

Keywords: Filter, Pruning, PCA, Depth separable convolution

1 Introduction

In recent years, deep learning technology has been developed rapidly, such as AlexNet [1], VGG [2], GoogleNet [3], ResNet [4] and other classic deep convolutional neural network architectures in image recognition, target detection, image classification, etc. It has been widely used in the field of computer vision. However, with the improvement of network performance and complexity, the number of layers, the number of parameters, and the amount of calculation [5] of high-precision network models are also increasing. At the same time, there is greater redundancy, which makes them difficult to be fully deployed on mobile and embedded devices with limited resources. In deep convolutional neural networks, the convolutional layer occupies most of the calculations, and the huge calculation consumes a lot of hardware resources such as CPU and GPU. Therefore, model compression has been developed rapidly. While ensuring that the accuracy of the network models is basically unchanged or slightly improved, the network model is compressed as much as possible to effectively reduce the amount of storage and calculation rate. Currently, network pruning [6], as an important research direction in network compression and acceleration, has received extensive attention from researchers at home and abroad.

At the earliest, Han et al. [7] proposed iterative pruning. The idea is to continuously prune the neural network after training the convergence to obtain a streamlined network model. Li et al. [8] evaluated the importance of each filter by calculating the L1 norm of each layer of filters. In addition, Liu et al. [9] added the sparse term of the scaling value of each layer to the loss function, and tailored the filter of each layer according to a given threshold. Yang He et al. [10] proposed to use soft filter pruning to accelerate the inference process of deep convolutional neural networks. When this method is trained after pruning, the convolution kernel that was pruned in the previous epoch is trained in the current epoch. When still participating in the iteration, those convolution kernels will not be directly discarded, maintaining the performance of the model to a large extent. On this basis, Yang He et al. [11] proposed a new filter pruning method called geometric median filter pruning (FPGM). FPGM selects the filter with the greatest contribution to replaceability. According to the geometric median (GM) characteristics of the same layer of filter in the model, the filters nearby can be expressed as the remaining filters. Although these methods ultimately result in a streamlined network with improved accuracy, as the depth of the network model increases, there are large computational and time costs, complex calculations, longer training time, and memory problems such as high consumption and high requirements for experimental hardware equipment.

Based on this, this paper proposes a filter pruning based on PCA and deep separable convolution to accelerate the inference process of deep convolutional neural networks.

*Corresponding Author: Fang-Chuan Ou Yang; E-mail: ouyang@ntub.edu.tw
DOI: 10.53106/160792642024032502014

First, the deep separable convolution is adopted to reduce the number of parameters needed for convolution calculation and thus reduce the amount of calculation. Second, by PCA dimension reduction, first of all, to distinguish the stacked similar filters, it can not only relieve the dimension disaster but also in the compressed data at the same time let the minimum information loss, the high-dimensional data into low-dimensional, so as to realize computation cost, time-consuming, and the reduction of storage, the classification model accuracy is improved. The results show that, compared with the above pruning method, the method proposed in this paper performs better on the Resnet after the Color Perception Test Chart training. It not only improves the accuracy of model pruning, but also greatly reduces the memory consumption of the terminal device.

The rest of this article is organized as follows: the second part explains the technical basis of deep convolutional neural networks, the third part introduces the method proposed in this article, the fourth part analyzes the experimental results, and the fifth part is the conclusion and future work.

2 Basics of Deep Convolutional Neural Network Technology

Deep convolutional neural networks have become a very common technical means in the field of computer vision and artificial intelligence applications [10], and significant technical results have been achieved in image classification, recognition, and target detection and tracking. However, most convolutional neural networks are computationally and storage-intensive, and it is difficult to deploy on mobile platforms and other micro-devices [10]. Therefore, it is particularly important to compress and accelerate the network model and reduce the computational load and storage space of the convolutional neural network. The size of the model depends on both the number of model parameters and the data types of the parameters. To further compress the model, current popular methods can be roughly categorized as follows: lightweight model [12] design, weights of quantitative [13], BN layer merging, network pruning, tensor decomposition, knowledge distillation, etc. Lightweight network model design mainly refers to considering some lightweight ideas at the beginning of model design, such as commonly used grouping convolution, deep separable convolution, 1*1 convolution to achieve channel dimensionality reduction, etc.

2.1 Ordinary Convolution and Depth Separable Convolution

Convolutional neural networks [14] not only have the advantages of traditional neural networks, but also have the characteristics of weight sharing, which gives convolutional neural networks significant advantages in image recognition. For convolution calculations, the convolution kernel can be regarded as a three-dimensional filter, consisting of the depth (namely, channel) dimension and the spatial dimension (including width and height). An ordinary convolution operation is to realize the joint mapping of depth correlation and spatial correlation, that is, both spatial information [14] and channel correlation need to be taken into account simultaneously, and then the output is non-linear activated. The depth separable convolution proposed on this basis [15] posits that the correlation between the depth of the convolutional layer and the spatial correlation can be decoupled. By mapping them separately, more features of the image can be captured, leading to better results.

2.1.1 Conventional Convolution

The conventional convolution operation process is shown in Figure 1, begins by entering a 32×32 pixel, three-channel (32×32×3) image, and passing it through the convolution layer of the 3×3 convolution kernel (assuming the number of output channels is 4), the convolution kernel is 3×3×3×4, and finally outputs 4 feature maps. If same padding is used, the output layer size is the same as the input layer (32×32). If not, the size becomes 3×3.

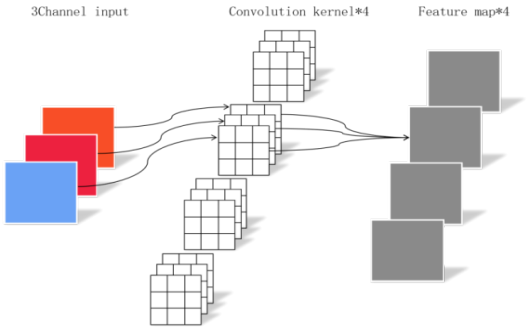

Figure 1. Traditional convolution process

2.1.2 Depth Separable Convolution

The depth separable convolution operation includes two parts: depthwise convolution and pointwise convolution.

Depthwise Convolution: Each convolution kernel is responsible for one channel, and each channel is convolved by only one convolution kernel. For the same 32×32 pixel, three-channel color input picture (32×32×3), the depthwise convolution first undergoes the first convolution operation, which is performed entirely in a two-dimensional plane. The number of convolution kernels corresponds to the number of channels in the previous layer (each channel corresponds to one convolution kernel). Therefore, a three-channel image is processed to generate 3 feature maps. If same padding is used, the size of the feature maps is the same as the input layer, which is 32×32, as shown in Figure 2:

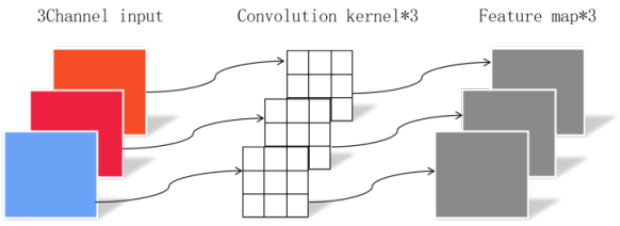

Figure 2. Channel-by-channel convolution process

The number of feature maps after the completion of channel-by-channel convolution is the same as the number of channels in the input layer, so the feature map cannot be

extended. Moreover, this operation convolves each channel in the input layer independently and fails to effectively utilize the feature information from different channels at the same spatial location. Therefore, point-by-point convolution is needed to combine these feature graphs to generate new feature graphs.

Pointwise Convolution: The operation of pointwise convolution is very similar to that of conventional convolution. The size of its convolution kernel is 1×1×M, where M is the number of channels in the upper layer. Therefore, the convolution operation here will weight and combine the feature maps from previous step in the depth direction to generate a new feature map. There are as many output feature maps as there are convolution kernels, as shown in Figure 3:

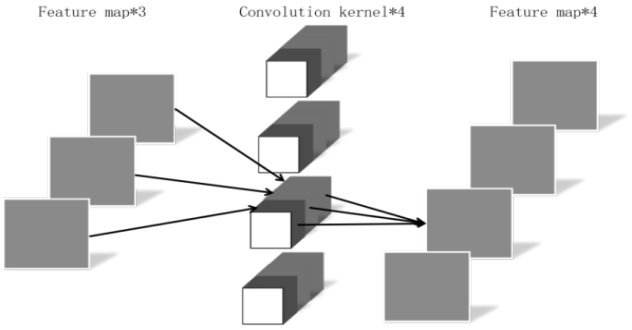

Figure 3. Point-by-point convolution process

2.2 Principal Component Analysis

PCA (Principal Component Analysis) is a common data analysis method often used for dimensionality reduction of high-dimensional data. Before the advent of deep learning technology, it was widely used in data feature extraction. Due to its excellent performance, it has been widely used in areas such as image face recognition and other aspects [16]. As we all know, an image often contains a large amount of information and many features during image processing. However, some information may not be very useful or may be relatively repetitive when performing image tasks. This provides us with ideas that an algorithm can be used to remove this kind of redundant information and extract the most important features. This is the purpose of the PCA method.

Here, we take the image [16] as an example to illustrate the implementation of the PCA method. There are mainly the following steps:

(1) Calculate the covariance matrix Cov of all images;
(2) Calculate the eigenvalues and eigenvectors of the covariance matrix Cov from step 1;
(3) Arrange the eigenvalues obtained in step 2 in descending order, and arrange the eigenvectors corresponding to the eigenvalues in the same order to form a transformation matrix;
(4) Perform a matrix multiplication operation with the original image using the transformation matrix obtained in step 3. The result is the matrix processed by PCA.

3 Filter Pruning Method Based on PCA Depth Separable Convolution

With the development of deep neural networks, the convolutional layers and BN layers are widely used in convolutional neural networks. In neural networks, most of the calculation operations are mainly concentrated in the convolutional layer, so reducing the calculation of the convolutional layer in the network structure can effectively reduce the amount of network parameters and calculations and accelerate the speed of network inference. In addition, for a deep convolution neural network, in the process of the calculation of similarity, the calculation of each layer of geometric median filter is a great workload, and in the process of the calculation, the similar filters are stacked together, forming a high-dimensional filter matrix, which is not only indistinguishable, but also results in dimension disaster. To solve these problems, this paper proposes a filter pruning method based on PCA dimensionality reduction and depth separable convolution.

3.1 Depth Separable Convolution

This paper replaces the conventional convolution part of the ResNet structure with a depth separable convolution [17], and its structure is shown in Figure 4. First, it performs a depthwise convolution [15], which separates the ordinary convolution in the spatial dimension to increase the network width and enrich the extracted features, and then it performs the pointwise convolution, which reduces the computational complexity of the convolution operation. In addition, the parameter quantity will not have a great impact on the accuracy of the experimental results [18].

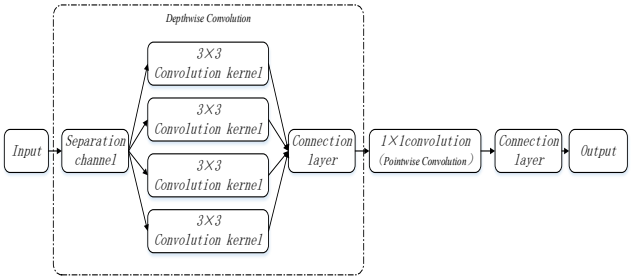

Figure 4. Depth separable convolution structure

The formula of depthwise convolution operation [15] is shown in formula (1), and the formula of pointwise convolution operation is shown in formula (2). Finally, Equation (1) is substituted into Equation (2) to obtain the depth separable convolution formula as shown in Equation (3).

$$\text{Depthwise Conv}\left(\mathrm{K},x\right)_{(i,j)} = \sum_{(k,l)} \mathrm{K}_{(k,l)} \otimes x_{(i+k,j+l)}. \quad \textbf{(1)}$$

In formula (1), K is the convolution kernel, x is the input feature map, i and j are the pixel positions of the feature map,

and k and l are the resolution of the output feature map, ⊗ which means that the corresponding elements are multiplied.

$$\text{Point wise}Conv(\mathrm{K},x)_{(i,j)}=\sum_{\mathrm{m}}\mathrm{K}_m\bullet x_{(i,j,m)}. \tag{2}$$

In formula (2), m is the number of channels.

$$\begin{aligned}&\mathrm{SepConv}(\mathrm{K}_p,\mathrm{K}_d,x)_{(i,j,m)}=\\&PonitwiseConv_{(i,j,m)}(\mathrm{K}_p,DepthwiseConv(\mathrm{K}_d,x)_{(i.j)}).\end{aligned} \tag{3}$$

In formula (3), Kd is the convolution kernel of depthwise convolution, and Kp is the convolution kernel of pointwise convolution.

In the network model training [19], considering the computational cost of the depth separable convolution, first, for the conventional convolution, we input the image of the M channel size of DI×DI, and after the convolution operation of the DK×DK we size convolution kernel, the output which is DO×DO N-channel feature map, and its calculation cost is shown in equation (4).

$$\cos t_{conv}=D_I\times D_I\times M\times N\times D_K\times D_K. \tag{4}$$

Then for the depth separable convolution, it is easy to know that the calculation costs of depthwise convolution and pointwise convolution are shown in equations (5) and (6), respectively.

$$\cos t_{depth}=D_I\times D_I\times M\times D_K\times D_K. \tag{5}$$

$$\cos t_{point}=M\times N\times D_I\times D_I. \tag{6}$$

Therefore, the computational cost of the total depth separable convolution operation is shown in equation (7).

$$\begin{aligned}&\cos t_{sep}=\\&D_I\times D_I\times M\times D_K\times D_K+M\times N\times D_I\times D_I.\end{aligned} \tag{7}$$

The computational cost comparison between depth separable convolution and conventional convolution [19] is shown in equation (8).

$$\begin{aligned}\frac{\cos t_{sep}}{\cos t_{conv}}&=\frac{D_I\times D_I\times M\times D_K\times D_K+M\times N\times D_I\times D_I}{D_I\times D_I\times M\times N\times D_K\times D_K}\\&=\frac{1}{N}+\frac{1}{{D_K}^2}.\end{aligned} \tag{8}$$

For convolutional neural networks in general, the size of the convolution kernel DK is usually 3×3, and the number of output channels N is typically 16, 64, 128, 256, 512, 1024. It can be seen that a conventional convolution can be replaced by the depth separable convolution, reducing the calculation cost to about 1/9 of the original, with only a small decrease in accuracy. Therefore, given the same number of parameters, the number of neural network layers with depth separable convolution can be made deeper.

3.2 PCA Dimensionality Reduction Accelerated Filter Pruning

In order to solve the calculation of geometric center of the increase of storage and computing cost, this paper uses PCA to reduce the dimensionality of the similar filter matrix obtained by calculating the geometric center. The purpose is to distinguish the stacked similar filters. It can alleviate the disaster of dimensionality and minimize the loss of information while compressing data, that is, converting high-dimensional data to low-dimensional, so as to achieve a significant reduction in computational cost, time-consuming, and storage, making the network model after pruning the effect better.

Assuming that there are m index variables for principal component analysis, there are n samples in total, and each sample is represented by the observation value of these m indicators, and an n×m sample data matrix [20] is obtained, as shown in equation (9):

$$X=\begin{bmatrix}x_{11}&\cdots&x_{1\mathrm{m}}\\\vdots&\ddots&\vdots\\x_{n1}&\cdots&x_{nm}\end{bmatrix}=\left[x_1,x_2,\cdots,x_m\right]. \tag{9}$$

Among them, xi = (x1i, x2i, ..., xmi)T, i = 1, 2, ..., m.

It can be seen that the covariance matrix of these m indicators is shown in equation (10):

$$\mathrm{D}_{m\times m}=Cov(X)_{m\times m}=E\left(X-EX\right)\left(X-EX\right)^T. \tag{10}$$

The PCA method aims to integrate the original m indicators x1, x2, x3, x4, ..., xm to form m uncorrelated new indicators, and obtain the following linear combination:

$$\begin{cases}y_1=a_{11}x_1+a_{21}x_2+\ldots+a_{m1}x_m\\y_2=a_{12}x_1+a_{22}x_2+\ldots+a_{m2}x_m\\\quad\ldots\\y_m=a_{1m}x_1+a_{2m}x_2+\ldots+a_{mm}x_m\end{cases}. \tag{11}$$

Can be abbreviated as:

$$y_i=Xa_i=a_{1i}x_1+\ldots+a_{mi}x_m,\ i=1,2,\ldots,m. \tag{12}$$

Where ai = (a1i, a2i, ..., ami)T, i = 1, 2, ..., m.

When the coefficient ai meets the following three conditions, the m comprehensive indicators obtained by the above transformation are not correlated with each other, and the variance decreases [21]:

(1) Yi and Yj (I ≠ j; I, j = 1, 2, ..., m) irrelevant;
(2) aiTai = 1, i = 1, 2, ..., m;
(3) The variance of y1 is the largest, and the variances of y2, ..., ym decrease sequentially.

At this time, y1, y2, ..., ym are respectively called the 1, 2, ..., m principal components of x1, x2, ..., xm. In practical applications, if the previous p(p < m) principal components are sufficient to reflect the information of the original m indicators to the greatest extent, and the first P principal components y1, y2, ..., yp are directly used to replace the original m indicators for subsequent analysis to achieve the purpose of dimensionality reduction.

The coefficients of the first p principal components in equation (11) are ai(i = 1, 2, ..., p) is the first p-large eigenvalue of the covariance matrix in equation (10) λi(i = 1, 2, ..., p), that is, the eigenvector corresponding to λ1 ≥ λ2 ≥ ... ≥ λp [22].

In order to obtain the p-order dimensionality reduction matrix of the training data set X, we take the first p feature vectors a1, a2, ..., ap to form the dimensionality reduction transformation matrix A = (a1i, a2i, ..., ami)T, i = 1, 2, ..., p, then the dimension reduction matrix Y after X dimension reduction is shown in formula (13).

$$Y_{n\times p} = X_{n\times m} A_{m\times p}. \tag{13}$$

In the practical application of PCA, the selection of the number of principal components P is very important. If the p value is too large, it may lead to too much noise information, but if the p value is too small, important information will be easily lost. Both of these situations are not conducive to the subsequent analysis. In order to determine the number of principal components well, the common method is to calculate the comprehensive evaluation value of principal components.

First, calculate the information contribution rate and cumulative contribution rate of the eigenvalue λi (i = 1, 2, ..., p). The information contribution rate formula of the principal component yi is shown in formula (14):

$$b_i = \frac{\lambda_i}{\sum_{k}^{m} \lambda_k} \left(i = 1, 2, \cdots, m\right). \tag{14}$$

$$b_i = \frac{\lambda_i}{\sum_{k}^{m} \lambda_k} \left(i = 1, 2, \cdots, m\right). \tag{15}$$

Formula (15) is the cumulative contribution rate of the main components y1, y2, ..., yp. When αp is close to 1 (αp = 0.85, 0.90, 0.95), the first p index variables y1, y2, ..., yp is used as p principal components to replace the original m index variables, so that p principal components can be comprehensively analyzed.

Finally, calculate the comprehensive score of each principal component, the formula is as shown in formula (16):

$$Z = \sum_{i=1}^{p} b_i y_i. \tag{16}$$

Among them, bi is the information contribution rate of the i-th principal component, which can be evaluated based on the comprehensive score value.

3.3 Cosine Similarity

Cosine similarity is different from Euclidean distance. It evaluates the similarity of two vectors by calculating the cosine of the angle between them. The formula is shown in formula (17):

$$similarity = \cos(\theta) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} (X_i \times Y_i)}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \times \sqrt{\sum_{i=1}^{n} (Y_i)^2}}. \tag{17}$$

Among them, X and Y are n-dimensional vectors, and Xi and Yi are the components of the vectors X and Y.

The similarity ranging from -1 to 1: -1 means that the two vectors point in the opposite direction, and 1 means that their directions are the same, while 0 usually means that the two vectors are independent. Therefore, the closer the cosine value is to 1, the closer the angle is to 0 degrees, which means the more similar the two vectors are.

3.4 Algorithm Flow Chart and Pseudocode

3.4.1 Algorithm Flowchart

Figure 5 is a flowchart of the PCA-based depth separable convolution filter pruning algorithm. The left half of the flowchart describes the process of replacing the first layer of conventional convolution in the ResNet network model with a depth separable convolution. Depth separable convolution consists of two stages: depthwise convolution and pointwise convolution. Depthwise convolution is the filtering stage of deep separable convolution. Each convolution kernel of the depthwise convolution layer only convolves with each input channel, while the pointwise convolution layer is responsible for different channels of the previous layer output. The linear combination of feature maps can effectively use the feature information of different channel feature maps at the same spatial position [23]. The right half is to perform PCA dimensionality reduction operation on the filter matrix obtained through similarity calculation to minimize the loss of information while compressing the data. The PCA-based depth separable convolution filter pruning algorithm improves the model convolution method of the FPGM [11] algorithm. The similar filter matrix obtained by the similarity calculation is reduced in dimensionality, which makes the

calculation cost and classification of the model classification The accuracy has been improved to some extent.

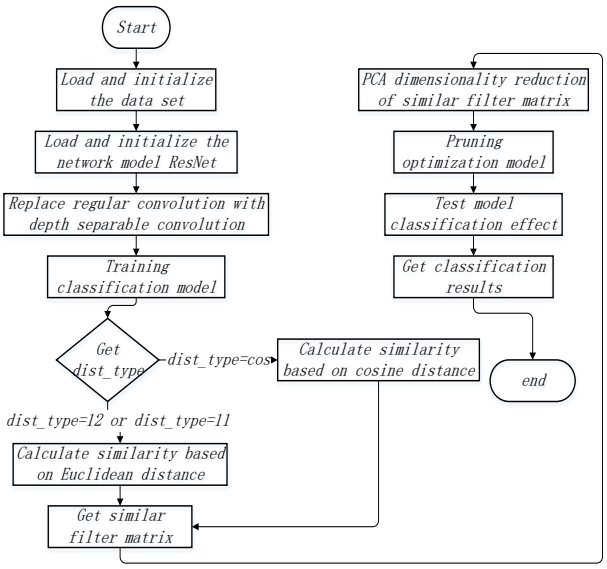

Figure 5. PCA-based depth separable convolution filter pruning algorithm flowchart

3.4.2 Pseudo Code

Algorithm 1. Algorithm description of PCA based depth separable convolution filter pruning

Input: training data: X.
1: **Given:** pruning rate Pi
2: **Initialize:** model parameter W={W(i), 0 ≤ i ≤ L}
3:**Instead:** torch.nn.Conv2d(groups=inplanes) AND outplanes=k*inplanes
3: **for** epoch = 1; epoch ≤ epoch-max; epoch++ do
4: Update the model parameter W based on X
5: for i = 1; i ≤ L; i + + do
6: if dist_type ==12 or dist_type==11:
7: Find $N_{i+1}P_i$ filters that satisfy Equation 4
8: else:
9: Find $N_{i+1}P_i$ filters that satisfy Equation 17
10: Get F_{i+1} filters that satisfy Equation 13
11: Zeroize selected filters
12: **end for**
13: **end for**
14: Obtain the compact model W* from W
Output: The compact model and its parameters W*

4 Experimental Results and Analysis

We designed and implemented a set of our own Color Perception Test Chart dataset generation algorithm for the experiment, which is based on the principle of vision and can be customized to control the number, size, shape, color and other attributes, and the diversity of the test graph can be improved by a randomized algorithm. The categories of the dataset contain numbers, letters, animals, etc., as shown in Figure 6. The dataset consists of 7,000 32x32 color images in 10 categories, of which 7/10 are training sets and 3/10 are test sets. The number of each category is equal, the training sets and test sets are subdivided into seven training batches and three test batch, each batch has 700 images.

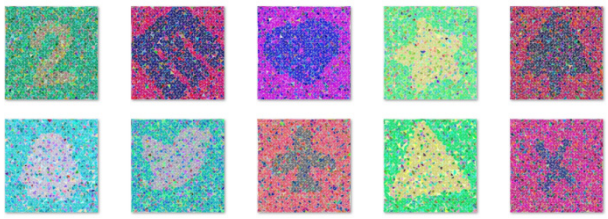

Figure 6. Example of a portion of the dataset before resizing

In an environment with Intel Core i7-9750h CPU, the open-source framework Torch was used to build the model. Torch is an open-source framework based on the BSD License for large-scale machine learning, particularly for image and video processing in the visual domain. Under the above experimental conditions, the method proposed in this paper was verified using the RESNET network model as the benchmark.

To evaluate the effect of this method on the different depth of ResNet, experiments were conducted at pruning rates of 40% to obtain the accuracy and loss changes of training and testing for ResNet 20, ResNet 32, and ResNet 56 on the Color Perception Test Chart. The change of accuracy is shown in Figure 7.

Figure 7. The accuracy rate changes of resnet-20, 32, 56 on the Color Perception Test Chart

When the pruning rates of ResNet20, ResNet32 and ResNet56 are 0.4, the loss changes of model classification are shown in Figure 8. It can be observed that the loss of the model is continuously declining, indicating that the method proposed in this paper can effectively enhance the accuracy of model classification.

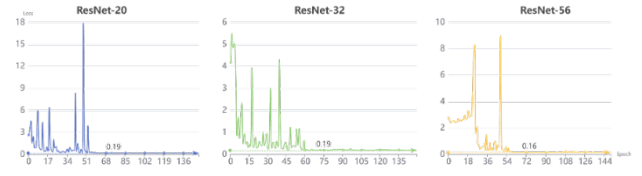

Figure 8. Loss changes of resnet-20, 32, 56 on the Color Perception Test Chart

Table 1 presents a comparison of the accuracy of model classification effects after applying different pruning criteria to the Color Perception Test Chart dataset for the residual networks of different depths. Table 2 compares the number of parameters before and after using PCA dimensionality reduction and depthwise separable convolution for residual networks of different depths. As shown in Table 1, when

the depth is 20,32,56 and pruning rate is 40%, the proposed method shows a certain improvement in model classification accuracy compared to other pruning strategies. The change in accuracy is depicted in Figure 9, which aligns with the expectations of this experiment. As indicated in Table 2, after applying the PCA dimensionality reduction and depthwise separable convolution methods proposed in this paper, the number of parameters of models with different depths was significantly reduced. A comparison of the MACs and parameters on residual networks is shown in Figure 10. In this experiment, the proposed method effectively balances the optimization and acceleration of accuracy, number of parameters, calculation amount, and running time. Given the significant fluctuations in model performance when training on the Color Perception Test Chart, we saved the best model as the model parameters obtained by training.

Through the experimental results, it can be observed that in the shallower network model, the method proposed in this paper can achieve better model acceleration effect. When the parameter amount is reduced by about 91%, the classification accuracy of the ResNet network model with depths of 20, 32, and 56 can still maintain a good accuracy after pruning. Furthermore, the running time and computational cost of the model are greatly reduced. However, the method proposed in this paper also has certain limitations. Inferring the pruned model without zeros function requires separate code and is not compatible with all networks.

Table 1. Accuracy comparison

Depth	Method	Acc. (%)	Acc. ↑
20	Baseline	95.42	0.00
	FPGM [11]-only 40%	94.72	-0.70
	FPGM [11]-DSC 40%	93.32	-2.10
	SFP [10]-only 40%	94.53	-0.89
	SFP [10]-DSC 40%	93.18	-2.24
	Ours-only 40%	95.79	0.37
	Ours-DSC 40%	94.44	-0.98
32	Baseline	96.17	0.00
	FPGM [11]-only 40%	93.97	-2.20
	FPGM [11]-DSC 40%	93.50	-2.67
	SFP [10]-only 40%	93.79	-2.38
	SFP [10]-DSC 40%	93.64	-2.53
	Ours-only 40%	95.98	-0.19
	Ours-DSC 40%	94.81	-1.36
56	Baseline	96.73	0.00
	FPGM [11]-only 40%	94.16	-2.57
	FPGM [11]-DSC 40%	93.93	-2.80
	SFP [10]-only 40%	93.93	-2.80
	SFP [10]-DSC 40%	93.69	-3.04
	Ours-only 40%	96.64	-0.09
	Ours-DSC 40%	95.05	-1.68

Table 2. Comparison of MACs and parameters

Depth	Method	MACs (M)	MACs↓ (%)	Params (K)	Params↓ (%)
20	Baseline	40.55	0.00	269.72	0.00
	FPGM [11]-only 40%	24.33	40.00	162.64	39.70
	FPGM [11]-DSC 40%	3.62	91.08	23.26	91.38
	SFP [10]-only 40%	24.33	40.00	162.64	39.70
	SFP [10]-DSC 40%	3.62	91.08	23.26	91.38
	Ours-only 40%	24.33	40.00	162.64	39.70
	Ours-DSC 40%	3.62	91.08	23.26	91.38
32	Baseline	68.86	0.00	464.15	0.00
	FPGM [11]-only 40%	41.32	40.00	279.66	39.75
	FPGM [11]-DSC 40%	6.12	91.11	39.48	91.50
	SFP [10]-only 40%	41.32	40.00	279.66	39.75
	SFP [10]-DSC 40%	6.12	91.11	39.48	91.50
	Ours-only 40%	41.32	40.00	279.66	39.75
	Ours-DSC 40%	6.12	91.11	39.48	91.50
56	Baseline	125.49	0.00	853.02	0.00
	FPGM [11]-only 40%	75.29	40.00	513.70	39.78
	FPGM [11]-DSC 40%	11.14	91.12	71.91	91.57
	SFP [10]-only 40%	75.29	40.00	513.70	39.78
	SFP [10]-DSC 40%	11.14	91.12	71.91	91.57
	Ours-only 40%	75.29	40.00	513.70	39.78
	Ours-DSC 40%	11.14	91.12	71.91	91.57

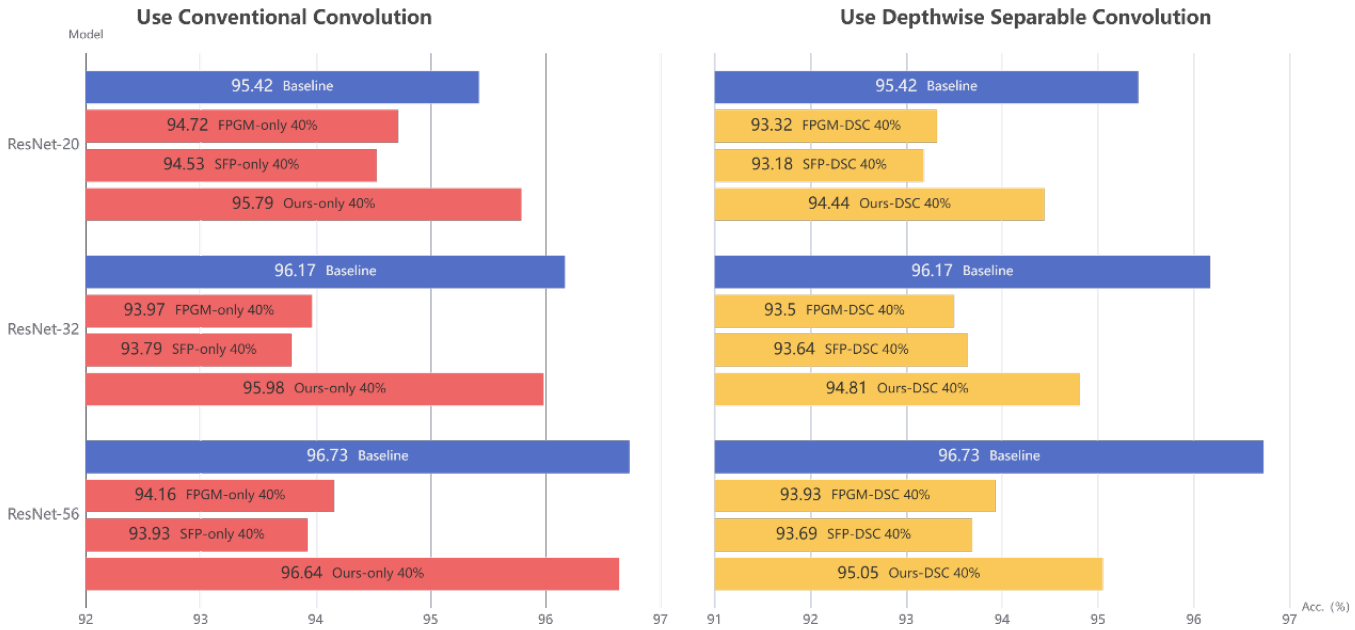

Figure 9. The pruning performance of different pruning criteria on different depth residual networks

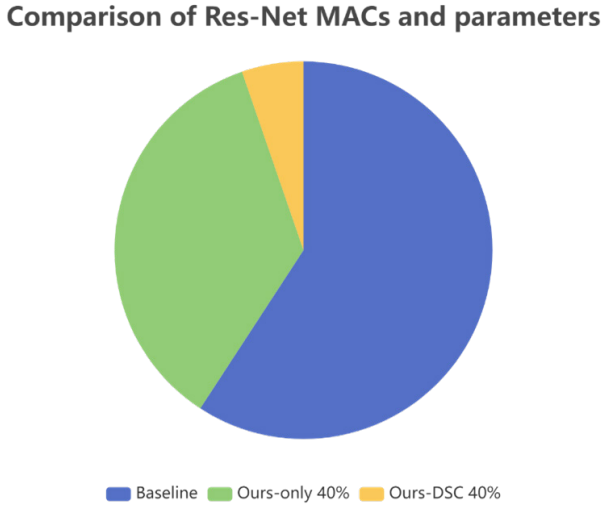

Figure 10. Comparison of MACs and parameters on residual networks

5 Summary and Prospect

In order to further compress the model, on the basis of filter pruning, in view of the large amount of calculation and dimensional disaster in deep convolutional neural networks, the main contribution of this algorithm is summarized as follows:

First, this article employs depthwise separable convolution instead of ordinary convolution in ResNet, and uses channel convolution to separate ordinary convolution in spatial dimensions to increase the network width and expand the range of feature extraction. Subsequently, pointwise convolution is used to reduce the ordinary the computational complexity of the convolution operation. This reduces the parameters required for the convolution calculation by about 90%, thereby decreasing the computational load.

Second, the use of PCA dimensionality reduction to distinguish the stacked similar filters can not only alleviate the dimensional disaster, but also achieve the effect of compressing data and minimizing information loss. In ResNet of different depths, the accuracy drop caused by pruning is relieved to a certain extent, and some results even show an accuracy increase of about 1%.

The method proposed in this paper significantly reduces the number of parameters in the convolution operation and enhances the performance of model classification. Building upon the filter pruning compression model, the model is further compressed and accelerated. However, when the number of network layers reaches a certain level, the accuracy will gradually decrease Therefore, this method is primarily suited for optimizing smaller network models. Future research will focus on optimizing deep neural networks, with further compression being pursued in the direction of pruning and quantization.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Proceedings of Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, 2012, pp. 1097-1105.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Proceedings of International Conference on Learning Representations*, San Diego, CA, USA, 2015, pp.1-14.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 1-9.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, 2016, pp. 770-778.
[5] Z. Xiao, X. Yang, X. Wei, X. Tang, Improved Lightweight Network in Image Recognition, *Journal of Frontiers of Computer Science and Technology*, Vol. 15, No. 4, pp. 743-753, April, 2021.
[6] L. Liu, S. Amirgholipour, J. Jiang, W. Jia, M. Zeibots, X. He, Performance-enhancing network pruning for crowd counting, *Neurocomputing*, Vol. 360, pp. 246-253, September, 2019.
[7] S. Han, J. Pool, J. Tran, W. J. Dally, Learning both weights and connections for efficient neural network, *Proceedings of Advances in Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 1135-1143.
[8] H. Li, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, Pruning filters for efficient convnets, *Proceedings of International Conference on Learning Representations*, Toulon, France, 2017, pp. 71-73.
[9] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, *Proceedings of IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2736-2744.
[10] Y. He, P. Liu, Z. Wang, Z. Hu, Y. Yang, Filter pruning via geometric median for deep convolutional neural networks acceleration, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 4340-4349.
[11] Y. He, X. Dong, G. Kang, Y. Fu, C. Yan, Y. Yang, Asymptotic soft filter pruning for deep convolutional neural networks, *IEEE Transactions on Cybernetics*, Vol. 50, No. 8, pp. 3594-3604, August, 2020.
[12] Q. Jia, *Research and implementation on deep convolutional neural network compression algorithm*, Master Thesis, Beijing Jiaotong University, Beijing, China, 2019.
[13] X. Jiang, Y. Xin, W. Liu, Vehicle detection model based on compressed and lightweight deep neural network, *Information Technology*, Vol. 44, No. 7, pp. 23-27, July, 2020.
[14] D. Wan, *Research on accelerating algorithm of neural network based on quantization*, Master Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2020.
[15] Y. Liu, X. Fu, X. Fu, W. Zhou, Z. Pan, Application of depthwise separable CNN in facial expression recognition, *Industrial Control Computer*, Vol. 33, No. 10, pp. 71-73, October, 2020.
[16] Y. Cao, L. Gui, Design of lightweight temporal convolutional network based on depthwise separable convolution, *Computer Engineering*, Vol. 46, No. 9, pp. 95-100, September, 2020.
[17] L. M. Kaiser, A. N. Gomez, F. Chollet, Depthwise separable convolutions for neural machine translation, *Proceedings of International Conference on Learning Representations*, Vancouver, Canada, 2018, pp. 1-14.
[18] B. D. Deebak, F. H. Memon, S. A. Khowaja, K. Dev, W. Wang, N. M. F. Qureshi, In the digital age of 5G networks: Seamless privacy-preserving authentication for cognitive-inspired internet of medical things, *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 12, pp. 8916-8923, December, 2022.
[19] Y. Xu, H. Lai, Z. Yu, S. Gao, Y. Wen, Chinese-Vietnamese neural machine translation based on depth separable convolution, *Journal of Xiamen University (Natural Science Edition)*, Vol. 59, No. 2, pp. 220-224, March, 2020.
[20] Z. Zheng, *Deep convolutional neural networks compression based on sparsity and quantization*, Master Thesis, Nanjing University of Information Science and Technology, Nanjing, China, 2020.
[21] D. Duan, *Research and application of feature reduction methods in text classification*, Master Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, 2020.
[22] P. Chen, *Research on principal component analysis and its application in feature extraction*, Master Thesis, Shaanxi Normal University, 2014. https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD201501&filename=1014400060.nh
[23] R. Guo, F. Wang, W. Liu, Vehicle type recognition based on improved depthwise separable convolution SSD, *Journal of Chongqing University*, Vol. 44, No. 6, pp. 43-48, June, 2021.

Biographies

Zheyi Wen entered Yangtze University in 2018, he will receive the bachelor's degree in computer Science and Technology from the university in 2022, his research interest covers computational intelligence, image and signal processing, and pattern recognition etc.

Chenlu Ye began her studies at Yangtze University in 2022, pursuing a master's degree in Business Administration. Her research interests encompass industrial and commercial big data management, as well as business intelligence decision-making.

Ming Zhao got Ph.D. in computer science and technology from Harbin Institute of Technology, China in 2015. He is currently a professor in Yangtze University, China. His research interests include computational intelligence, image and signal processing, pattern recognition etc. He is an IEEE Senior Member.

Fang-Chuan Ou Yang got Ph.D. from the Department of Information Management at National Central University in 2009. He is currently an Associate Professor in the Department of Digital Multimedia Design at National Taipei University of Business, Taiwan. His research interests include software engineering, artificial intelligence, AR/VR/MR, and educational robotics.