IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Artificial intelligence image recognition method based on convolutional neural network algorithm

Youhui Tian
Jiangsu Vocational Institute of Commerce; Nanjing Jiangsu,211168 ,China
Corresponding Author: Youhui Tian (E-mail: tianyouji123@126.com).

ABSTRACT As an algorithm with excellent performance, convolutional neural network has been widely used in the field of image processing and achieved good results by relying on its own local receptive fields, weight sharing, pooling, and sparse connections. In order to improve the convergence speed and recognition accuracy of the convolutional neural network algorithm, this paper proposes a new convolutional neural network algorithm. First, a recurrent neural network is introduced into the convolutional neural network, and the deep features of the image are learned in parallel using the convolutional neural network and the recurrent neural network. Secondly, according to the idea of ResNet's skip convolution layer, a new residual module ShortCut3-ResNet is constructed. Then, a dual optimization model is established to realize the integrated optimization of the convolution and full connection process. Finally, the effects of various parameters of the convolutional neural network on the network performance are analyzed through simulation experiments, and the optimal network parameters of the convolutional neural network are finally set. Experimental results show that the convolutional neural network algorithm proposed in this paper can learn the diverse features of the image, and improve the accuracy of feature extraction and image recognition ability of the convolutional neural network.

INDEX TERMS Convolutional neural network; Artificial intelligence; Image recognition.

I. INTRODUCTION

With the rapid development of the mobile Internet, the widespread use of smart phones and the popularization of social media self-media, a large amount of picture information has accompanied [1-2]. Nevertheless, as pictures become an important carrier of network information, problems also arise. Traditional information materials are recorded by words, and we can retrieve and process the required content by searching keywords. However, when pictures express the information, we cannot retrieve or process the information expressed in the pictures. The picture brings us a convenient way of information recording and sharing, but it is difficult for us to use the information expressed by the image. In this case, how to use a computer to intelligently classify and recognize the data of these images is particularly important [3-4].

In traditional pattern recognition methods, the most important thing is to express this image through a mathematical statistical model after extracting a certain amount of artificial feature points [5-6]. Then identify the image by the method of image matching. The basic principle of this method is that the similar samples are very close in the pattern space and form a "clustering", and then combined with the classifier for classification and recognition. For example, object recognition uses scale-invariant feature transform (SIFT) features, face recognition uses local binary patterns (LBP) features, and pedestrian detection uses histogram of oriented gradient (HOG) features, but such shallow machine learning methods have low recognition. With the development of artificial intelligence, continuous breakthroughs in deep learning have achieved great success in the fields of speech recognition, NLP processing, computer vision, video analysis, multimedia, and so on [7-9]. More and more enterprise companies and researchers use deep learning to discuss and study image classification, which provides a good development for artificial intelligence.

In previous years, most methods used shallow structural models to process data, and the structural model had at most one or two layers of nonlinear features. The most representative shallow structures are Gaussian Mixture Model [10], K-means clustering [11], Support Vector Machine [12], and Logistic Regression [13]. Convolutional neural network can extract the connection and spatial information

between its layers from the image, and can express the relevant characteristics inside the image [14-15]. The image recognition process based on deep learning is mainly to input the image into the neural network, and use the deep learning forward propagation and back propagation error algorithms to minimize the loss function. After updating the weights, a better recognition model is obtained. Then use this model to identify new images.

In practical applications, CNN has been used in many visual pattern recognition systems. Morvan et al. [16] proposed the CNN structure LeNet for handwritten digit recognition. Convolutional neural networks are also used for facial recognition and facial localization [17]. Parmar et al. [18] used convolutional neural networks to detect faces and facial expressions. Grossberg et al. [19] used shunt suppression convolutional neural networks for the detection of eye and face images. Nguyen et al. [20] used convolutional neural networks to detect text images. Wang et al. [21] achieved victory in the large-scale image recognition competition using the classic AlexNet model, and successfully reduced the false recognition rate to 17%. After the success of AlexNet, the researchers proposed other network models, such as VGGNet [22], GoogleNet [23], and ResNet [24]. AlexNet uses the ReLU function to replace the activation function commonly used in traditional neural networks [25]. Compared with sigmoid function and Tanh function, ReLU has no exponential calculation, the calculation amount is small and the network will not be saturated, and because of the linear unsaturated form of ReLU, it will speed up the network convergence speed. To solve the problem that the ReLU function is not derivable at the origin, Montanelli et al. [26] proposed a sparse ReLU function. Wang et al. [27] proposed parametric ReLU, and proved through experiments that the PReLU function has achieved good results in the big data classification task. In recent years, the study of convolutional neural networks has been inseparable from transfer learning. Transfer learning is a method that can use the knowledge that has been learned to solve problems in new fields [28]. Feng et al. [29] used large image datasets to perform pre-training on convolutional neural networks, and then trained and tested the trained networks on the image datasets to be classified. Compared with the traditional method of training the network directly on the target data set, the image recognition rate of this method is greatly improved. Zhang et al. [30] proposed an algorithm based on hierarchical sparse coding (HSC), which extracts features through hierarchical pooling and sparse processing, and obtains good results in handwriting recognition and multi-class object recognition. Carroll et al. [31] proposed a scattering convolutional neural network (Scatter-Net) based on wavelet transform, which uses wavelet transform to extract image high-frequency information hierarchically instead of the parameter learning process, which shows good performance in image recognition and classification tasks. Hu et al. [32] proposed a model PCAnet that initializes the CNN convolution layer parameters by extracting the features of the image principal component, and has achieved good results in image recognition tasks. Wang et al. [34] pointed out that the structure of the convolutional neural network itself is the main factor for the network to extract multi-level and multi-scale features. Sadr et al. [35] combined a convolutional neural network and a recurrent neural network to propose a new deep learning structure. The convolutional neural network learns the shallow features of the original image and uses it as the input of the recurrent neural network. Using the recurrent neural network to learn the high-level features, it achieves a good recognition rate in color-depth image recognition. Guo et al. [36] added the input of the convolutional neural network based on the reference [35], and proposed a multi-scale convolutional recurrent neural network. After local contrast normalization and sampling, it was used as the input of the recurrent neural network to extract more abstract high-level features.

Although there are many image recognition algorithms based on convolutional neural networks, and the recognition effect is very good. However, many recognition algorithms are now based on a specific database to design the depth and level of the network. Through continuous exploration, the best parameters and optimization algorithms are found. The human factor is relatively large, and there is no systematic theory to affect the recognition effect of the convolutional nerve. Especially when classifying and recognizing natural images, the selection of the initial state parameters of the convolutional neural network and the optimization algorithm will have a great impact on the network training. If the selection is not good, the network will not work, or it may fall into the local minimal, under-fitting, over-fitting, and many other problems [37-38].

In order to improve the ability of the convolutional neural network to classify and recognize two-dimensional images, speed up the convergence of the algorithm, reduce the number of iterations and shorten the training period, and achieve good classification results, this paper proposes a new convolutional neural network algorithm. First, a recurrent neural network is introduced into the convolutional neural network, and the deep features of the image are learned in parallel using the convolutional neural network and the recurrent neural network. Secondly, according to the idea of ResNet's skip convolution layer, a new residual module ShortCut3-ResNet is constructed. Finally, a dual optimization model is established to achieve integrated optimization of the convolution and full connection process.

Specifically, the technical contributions of our paper can be concluded as follows:

This paper proposes a new convolutional neural network algorithm. The network can learn the multi-scale and diverse features of the image. Moreover, without adding additional parameters and calculations, it can improve the accuracy and recognition ability of convolutional neural network feature extraction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

The rest of our paper was organized as follows. Related work was introduced in Section II. Section III described the structure of the convolutional neural network algorithm proposed in this paper. Experimental results and analysis were discussed in detail in Section IV. Finally, Section V concluded the whole paper.

II. Related works

A. Basic introduction to convolutional neural networks

In recent years, researchers have applied CNN to other fields, such as speech recognition [39], face recognition, object recognition, natural language processing [40], brain wave analysis [41], and so on. These fields continue in many directions and some breakthroughs have been made. Compared to ordinary neural networks, CNN contains a feature extractor, which consists of a convolution layer and a down-sampling layer. A neuron is only connected to a part of neurons in the upper layer, and this part of neurons is called a local receptive field. A convolutional layer generally contains multiple feature maps, each feature map is composed of a specific number of neurons, and the weights of neurons are shared between the same feature maps. The biggest feature of weight sharing is to reduce the connection between the various layers of the network, reduce network parameters, and at the same time play a role in preventing overfitting. In general, the initial value of the convolution kernel is randomly generated. In the process of network training, new weights are constantly learned and updated in real time until a reasonable weight is finally learned. Down sampling, also called pooling, is a special convolution process. Therefore, CNN has three main features, namely local receptive field, weight sharing and pooling [42].

(1) Local receptive field

In CNN, each neuron of the hidden layer is connected to a small area of the input layer, and each connection has a parameter weight and offset that can be learned. This area is called the local receptive field of the hidden layer neuron. Each neuron corresponds to a local receptive field.

(2) Weight sharing

For a local receptive field with 25 pixels, there is a weight of 5×5 for each neuron in the hidden layer. Weight sharing is that the weights corresponding to these neurons in the hidden layer are the same. Due to the existence of weight sharing, the amount of network parameters and training time are greatly reduced.

(3) Pooling

The pooling layer is generally behind the convolution layer, the purpose is to compress the image after the convolution to reduce the amount of parameters.

B. The structure of convolutional neural network

Convolutional neural network is a combination of deep learning algorithm and artificial neural network, and it is widely used in image processing. Convolutional neural network is generally composed of three parts: input layer, hidden layer and output layer. Among them, the input layer is the original image that has not been processed, the output layer is the result of classifying the features, and the hidden layer is a neuron layer with a complex multi-layer nonlinear structure, including a convolution layer and a sub-sampling layer. Convolutional neural networks extract and classify features in hidden layers. Therefore, optimization of the convolutional layer and single-layer perceptron can improve the accuracy of feature extraction and optimize the classification effect. The structure of the convolutional neural network is shown in Figure 1.

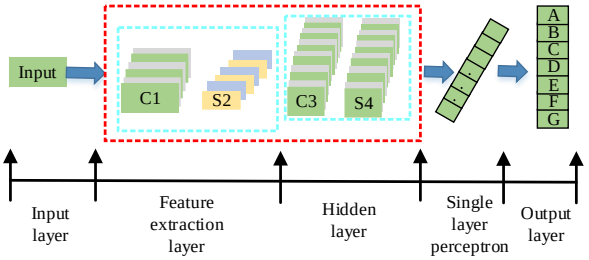

Figure 1. Structure of a convolutional neural network

Figure 1 shows the structure model of a convolutional neural network with only 2 convolutional layers (C1, C3) and 2 sub-sampling layers (S2, S4) in the hidden layer. The input data in the figure is the original image input, and the output result of the output layer is divided into A~G categories. The repeated structure composed of layer C and layer S serves as the basic unit of feature extraction. After multiple feature extractions, the final feature map is rasterized to obtain a one-dimensional matrix, which is a fully connected layer. The fully connected layer and the output layer use the fully connected method to obtain the final output result.

(1) Convolutional layer

Convolutional layers are the most hidden layers in CNN. At present, the structure of CNN has gradually tended to stack several convolutional layers in succession, followed by a pooling layer.

The function of convolution is to convolve the input image and the filter, the information of the original image is enhanced, and the interference of noise is suppressed. The process of convolution reflects the characteristics of local receptive field and weight sharing. In convolution, the network will automatically learn the features without manual selection of features, which avoids time and effort. Suppose there is an image with a size of 4×4, a 2×2 convolution kernel is used, and the sliding step of the convolution kernel is one, then the convolution operation process can be expressed as shown in Figure 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

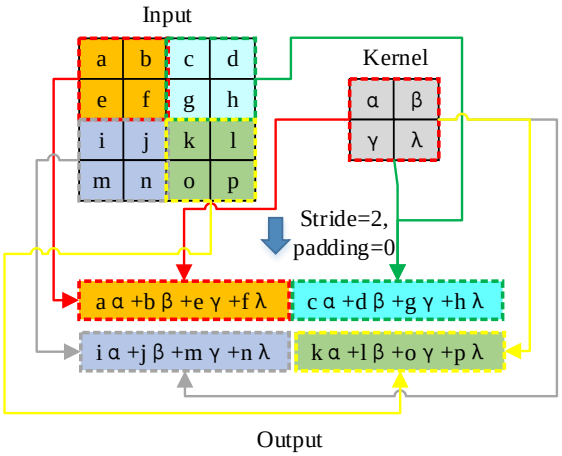

Figure 2. Process of convolution operation

(2) Pooling layer

The pooling layer is also a very common type of hidden layer used in CNN. Because the local features in the image are related, pooling the image can greatly reduce the amount of calculation but will not lose the main features of the image. Assuming that the size of the image is 4×4, a convolution kernel of 2×2 size is used, and the sliding step of the convolution kernel is set to two, the common pooling methods, maximum pooling, average pooling and random pooling the process is shown in Figure 3.

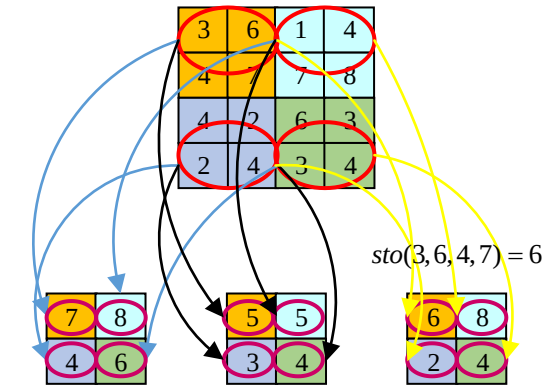

Figure 3. The process of pooling operation

(3) Active layer

Another important hidden layer of CNN is the active layer. When solving more complex problems, the activation function adds nonlinear factors into the neural network so that, the neural network can adapt to more problems that are complex. Commonly used activation functions include sigmoid function, Tanh function, ReLU function, and leaky ReLU function. The formula of sigmoid function is shown as formula (1). The Tanh function can be expressed as formula (2). The ReLU function is described by formula as shown in equation (3). The improvement of the gradient problem caused by the negative ReLU input results in the Leaky ReLU function, whose function formula can be expressed as formula (4).

$$\sigma(x) = 1/(1+e^{-x}) \quad (1)$$

$$Tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \quad (2)$$

$$f(x) = \max(0, x) \quad (3)$$

$$f(x) = \max(\alpha x, x) \quad (4)$$

The corresponding graphs of these four functions are shown in Figure 4 below.

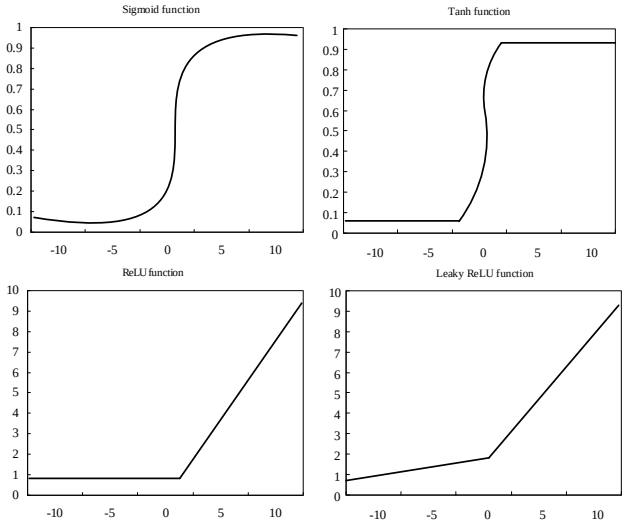

Figure 4. Curves of four activation functions

(4) Fully connected layer

The fully connected layer is the last few layers in the CNN and acts as a classifier in the entire network [43]. If the previous layer of the fully connected layer is also a fully connected layer, then 1×1 convolution is used. If the previous layer of the fully connected layer is a convolutional layer, the global convolution of $h \times w$ is used, and variable h and variable w are the height and width of the previous layer convolution result, respectively.

C. ResNet Introduction

Since AlexNet, some cutting-edge CNNs are constantly deepening [44-45]. For example, AlexNet has five convolutional layers; VGGNet and GoogleNet have 19 and 22 convolutional layers, respectively. However, it is not feasible to increase the network depth by directly superimposing layers. Assuming there is a shallow network, multiple maps are stacked on this network to form a deep network. In theory, the training error of this deep network will not be higher than that of the shallow network. However, the experimental results show that such a deep network cannot be found. After doing many experiments, it is also found that the deep network shows higher training error than the shallow network on the same data set, as shown in Figure 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

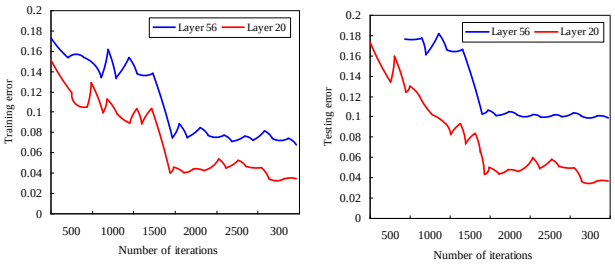

Figure 5. The error of the 20-layer and 56-layer networks on the CIFAR-10 dataset

In Figure 5, for the 20-layer and 56-layer networks, the experimental results on CIFAR-10 show that the error of the 56-layer network is higher than that of the 20-layer network. This phenomenon is called the degeneration problem.

Due to the problem of vanishing gradients, deep neural network training becomes very difficult. The gradient disappearance problem means that when the gradient is propagated back to the previous layer, repeated multiplication will make the gradient infinitely small. Therefore, as the network continues to deepen, its performance will gradually become saturated.

Therefore, drawing on the idea of cross-layer connections in high-speed networks, ResNet was proposed [46]. The core of the high-speed network is the addition of two nonlinear conversion layers to the ordinary neural network. One is T (transform gate) and one is C (carry gate), as shown in equation (5).

$$y = H(x, W_H) \square T(x, W_T) + x \square C(x, W_C) \quad (5)$$

Suppose that the goal of the neural network complex submodule is to learn H(x). If this target mapping is complicated, it is difficult for the neural network to learn from it. Then, the target mapping can be directly learned without the module. Instead, learn the difference of H(x) − x. This difference is called the residual, that is, F(x) = H x) − x. Therefore, the original target mapping is H(x) = F(x) + x, which constitutes ResNet. This cross-layer connection network structure breaks the convention that the output layer of n-1 traditional neural network can only give n layers, so that the output of a layer can directly pass through the problem of gradient disappearance or gradient explosion after several layers of input. The residual module can be expressed as formula (6) and formula (7).

$$y_i = F(x_i, W_i) + h(x_i) \quad (6)$$

$$x_{i+1} = f(y_i) \quad (7)$$

Among them, the variable x_i is the input vector. The variable x_{i+1} is the output vector. The variable F represents the residual mapping that the residual structure needs to learn and can be expressed as $F = W_2\sigma(W_1 x)$. The variable f represents the activation function operation. The variable $h(x_i) = x_i$ is a cross-layer connection.

Nevertheless, in CNN, as the network deepens, the number of convolution kernels also increases. Therefore, for the case where these two dimensions do not match, a special convolution kernel W_s can be used to perform the convolution operation to ensure the matching of the two dimensions.

$$y_i = F(x_i, W_i) + W_s h(x_i) \quad (8)$$

The combination of residual block and BN in ResNet can solve the gradient dispersion problem well. BN plays a normalized role in CNN, as shown in Figure 6. Neural networks need to learn how the input data is distributed. If the data used for the training network and the data used for the test network come from different distributions, the generalization ability of the network thus trained will be poor. Moreover, for the training data, if the distribution of each batch is not the same, the network needs to learn a different distribution, which will directly increase the training time of the network. In addition, for the activation function, the distribution of data is also very important. When the data distribution range is too large, the nonlinear characteristic of the activation function is not conducive to the utilization.

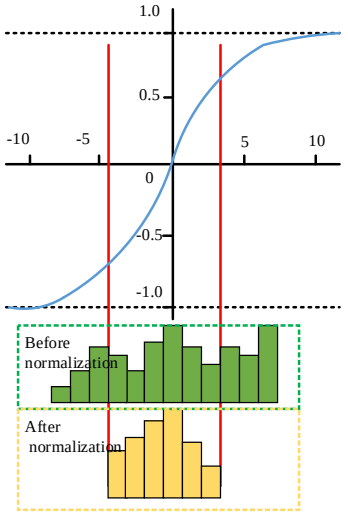

Figure 6. Schematic diagram of normalization

It can be seen from Figure 6 that many parts of the data before normalization may be located in the saturation region of the loss function, resulting in very close output results, which is not conducive to network learning. Adding the BN layer can solve this problem, that is, before the data is sent to the next layer, the data is normalized.

The forward propagation process of BN is as follows:

Assuming that the input data is xi, first calculate the mean value of the data, as shown in equation (9).

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad (9)$$

Among them, the variable m represents the size of mini-batch. The above formula calculates the average value of mini-batch. Then calculate the standard deviation of mini-batch as shown in equation (10).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \quad (10)$$

Then normalize the input data as shown in equation (11).

$$\hat{x}_i = \frac{(x_i - \mu_B)^2}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (11)$$

Then the output result of the BN layer can be obtained as shown in equation (12).

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma\beta}(x_i) \quad (12)$$

In the formula, the variable γ represents the scaling factor. The variable β represents the offset coefficient. The essence of the BN layer is to learn these two parameters, so that the network can learn the feature distribution of the output.

III. Image recognition algorithm based on CNN

This paper first introduces a recurrent neural network into the convolutional neural network, and uses the convolutional neural network and the recurrent neural network to learn the deep features of the image in parallel. Secondly, according to ResNet's idea of skipping convolutional layer, a new residual module ShortCut3-ResNet is constructed. Finally, a dual optimization model is established to achieve integrated optimization of the convolution and full connection process. Next, we will explain systematically.

A. Recurrent neural network

Recursive neural network is similar to the combination of convolution operation and sampling operation. By repeatedly using the same set of weights and selecting the acceptance domain to achieve the purpose of reducing the feature dimension layer by layer, its structure diagram is shown in Figure 7. Among them, variable K_1 is the number of feature maps output by the first-level network. The size of the bottommost feature map is 4×4, which is the unit of the bottom feature map. Let the acceptance field be 2×2 and the connection weight is W. Each unit of the second-layer network feature map is connected to the 2×2 acceptance field of the bottom layer feature map, and finally a 2×2 size feature map is obtained. In the same way, the second layer of feature maps gets a 1×1 size feature map after going through a layer of recurrent neural network.

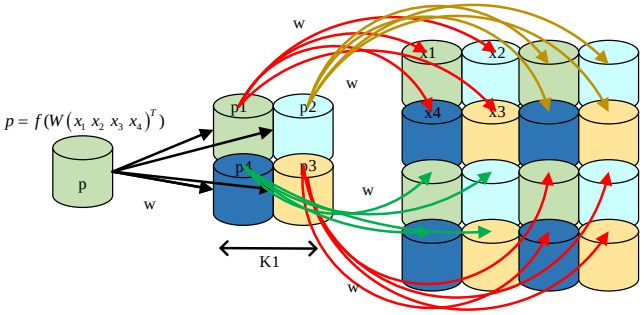

Figure 7. Schematic diagram of the structure of a recurrent neural network

B. Construct a new residual module

ShortCut in ResNet skips the two convolutional layers and connects to the corresponding output layer, as shown in Figure 8(a).

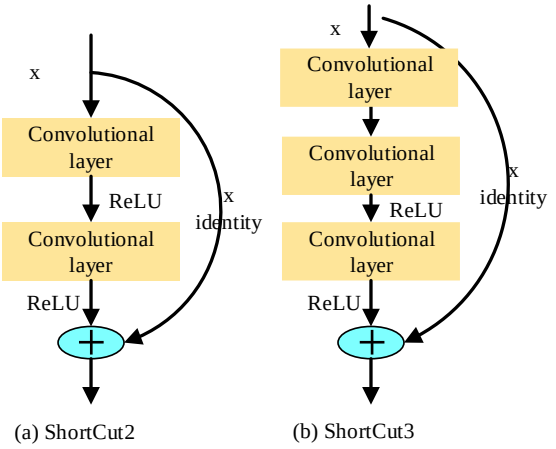

Figure 8. ShortCut in ResNet

According to ResNet's idea of skipping convolutional layers, this paper builds a new residual module that skips three convolutional layers, as shown in Figure 8(b). The resulting ResNet is called ShortCut3-ResNet. The most important part of ResNet is the shortcut. Although its presence makes the network look more complicated, it does not add additional parameters and calculations, but improves the accuracy of recognition.

In the design of ShortCut3-ResNet network, this paper draws on the design model of VGGNet. All the convolution kernels in the network adopt 3×3 convolution kernels, and can be divided into 3 segments according to the number of convolution kernels. Each segment has $2n$ layers (n>= 3). The number of convolution kernels in each layer in the first paragraph is 16. The number of convolution kernels in each layer in the second paragraph is 32. The number of convolution kernels in each layer in the third paragraph is 64. In summary, except for the first layer and the last layer, the entire network has $6n$ hidden layers.

Table 1. Two different ResNet structures

Convolutional layer	Number of convolution kernels	ResNet	ShortCut3-ResNet
Conv1	16	3×3, 16	3×3, 16
Conv2_x	16	$\begin{pmatrix}3\times3,\ 16\\3\times3,\ 16\end{pmatrix}\times3$	$\begin{pmatrix}3\times3,\ 16\\3\times3,\ 16\\3\times3,\ 16\end{pmatrix}\times2$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

Conv3_x	32	$\begin{pmatrix}3\times3,\ 32\\3\times3,\ 32\end{pmatrix}\times3$	$\begin{pmatrix}3\times3,\ 32\\3\times3,\ 32\\3\times3,\ 32\end{pmatrix}\times2$
Conv4_x	64	$\begin{pmatrix}3\times3,\ 64\\3\times3,\ 64\end{pmatrix}\times3$	$\begin{pmatrix}3\times3,\ 64\\3\times3,\ 64\\3\times3,\ 64\end{pmatrix}\times2$
1×1 Pooling			

In the network structure described above, the first layer uses a 3×3 convolutional layer. Nevertheless, the last layer no longer uses the fully connected layer in VGGNet, but borrows from the global average-pooling layer in network. This can effectively avoid the problems of excessive parameter quantity, low training speed, and easy overfitting in the fully connected layer.

When the value of n is three, we can get the number of hidden layers as 18, plus the first layer of convolutional layer and the last layer of global average pooling layer, we can get a 20-layer ShortCut3-ResNet. Table 1 describes the original ResNet and ShortCut3-ResNet network configurations for the 20-layer network. Based on this structure, $6n+2$ (n>=3) layer network structure can be obtained, as shown in Figure 9.

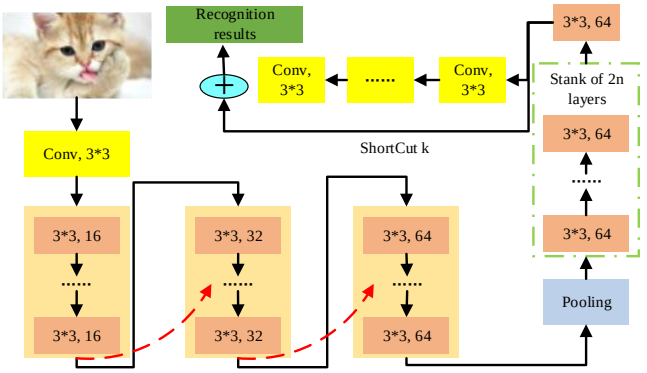

Figure 9. Schematic diagram of network structure

This network has three convolutions. According to the depth of the network in each convolution, there will be multiple convolutional layers. The number of convolution kernels in each segment is the same, and the later the number of convolution kernels increases. The connection of the dashed and solid lines of ShortCut in Figure 9 is different. Because the residual learning is H(x) = F(x) + x, it is the addition of the two channels of F(x) and x. The number of the two channels at the solid line connection is the same and can be added directly. The dotted line indicates that the two channels have different numbers, and the dimension of x needs to be adjusted by a convolution operation, that is, the convolution kernel W_s in formula (8).

C. Double optimization

The design principle of the convolution optimization model is to realize the weight optimization of the convolution kernel. We can learn the data set weights and bias parameters from small data blocks to obtain a sparse feature matrix. The convolution kernel is initialized by convolution coefficient control. Let matrix X is the sample data set. A is the base matrix used to transform X from sample space to feature space. Matrix S is the feature table of the data set. Setting the objective function $J(A,S)$ and assigning the initial value of S, the process of reducing the objective function through iteration is the process of optimizing S. Giving a good initial value of S can avoid the situation of poor convergence during the iteration process, and at the same time obtain a faster convergence and more optimized results. The process of S initialization and feature update is as follows:

$$S = G(W^T X) \quad (13)$$

$$S_c^{'} = S_c / \| A_c \| \quad (14)$$

Among them, the variable W^T is a random orthogonal matrix. Using W^T to extract sample X through weighted transformation, the initial value of matrix S is generated. The variable S_c represents the c-th feature matrix of the matrix S. The variable A_c represents the corresponding base matrix of S_c in matrix A. Let variable M be $m\times n$ matrix, then:

$$\| M \|_k = (\sum_{i=1}^{m}\sum_{j=1}^{n} | m_{ij} |^k)^{1/k} \quad (15)$$

This normalization process can maintain sparsity without affecting the performance of the algorithm. Therefore, through the above process, S can obtain a good initial value. The objective function expression is:

$$J(A,S) = \| AS - X \|_2^2 + \gamma \| A \|_2^2 \quad (16)$$

The variable $\| AS - X \|_2^2$ is the error between the samples set reconstructed using the base matrix and the feature set and the actual sample set. The variable $\gamma \| A \|_2^2$ is the sparse control term. The variable γ is the sparse coefficient. The value of variable $J(A,S)$ is the sum of the error term and the sparse control term.

Step 1 Given a random initialization matrix A.

Step 2 According to the given A and S, use the gradient descent method to find the local minimum value of the objective function $J(A,S)$. Moreover, obtain the value S 'of S at this time, α is the step size, and control the change amount of the gradient direction each time. The calculation process is:

$$S^{'} = S - \alpha \frac{\partial J(A,S)}{\partial S} \quad (17)$$

Step 3 Use the gradient descent method to obtain the local minimum value of the objective function $J(A,S)$ again according to the value of variable $S^{'}$. Moreover, obtain the value of variable $A^{'}$ at this time, α is the step

size, and control the change amount of gradient direction each time. The calculation process is

$$A^{'} = A - \alpha \frac{\partial J(A,S)}{\partial S} \quad (18)$$

Step 4 Use A 'and S' to replace A and S respectively, repeat step 2 ~ step 4.

As the number of iterations increases, the objective function will gradually decrease in the reverse direction of the gradient until the gradient vector approaches zero, and the objective function no longer decreases or the change can be ignored. Randomly sample the obtained feature matrix S to construct the initial weight of the convolution kernel. At the same time, the dynamically determined value μ is used in the convolutional neural network to replace the constant μ_0 as the convolution coefficient, to realize the optimization of the convolution kernel.

Suppose the convolutional neural network contains a total of k convolutional layers, the size of each convolution kernel is $l_{\ker} \times l_{\ker}$. The input image size of the convolutional layer is a matrix of variable $l_{img} \times l_{img}$. The input and output feature maps or images are n_{in} and n_{out} respectively.

Let the matrix Opt_1 be the feature matrix S when the objective function obtains the minimum value after multiple iterations. Use the convolution coefficient to optimize the convolution kernel, analyze the original convolution result through the dichotomy, and construct the function expression according to the interpolation principle. The dynamic convolution coefficient μ is expressed as follows

$$\mu = \frac{n_{in} n_{out} l_{img}^2}{2^k} + \theta_1 \quad (19)$$

Among them, the variable θ_1 is the correction error term. The expressions of the number of parameters required for the input data and output data corresponding to the convolution kernel are as follows:

$$f_{in} = n_{in} l_{\ker}^2 \quad (20)$$

$$f_{out} = n_{out} l_{\ker}^2 \quad (21)$$

The initialization expression of optimized convolution kernel is as follows:

$$Opt_2 = 2\sqrt{\frac{\mu}{f_{in} + f_{out}}} \times Opt_1 \quad (22)$$

The variable Opt_2 is the final optimized parameter matrix of convolution kernel. Suppose the convolutional neural network contains a total of k convolutional layers, and all input images are divided into n_{cag} categories. The number of iterations required for a convolutional neural network is ε. The number of feature maps generated by the last sub-sampling layer received by the single-layer perceptron is n_f.

The optimization process of fully connected parameters is similar to convolution optimization, and the process is as follows.

Let Opt_3 be a fully connected parameter matrix randomly initialized using parameters n_{cag} and $n_f l_{img}^2$. According to the operation process and classification results of the convolutional neural network, the parameter settings of the fully connected layer are affected by factors such as the number of iterations of the convolutional neural network. According to the interpolation principle, the constructor optimizes the parameters of the full connection, and ρ is the optimization coefficient:

$$\rho = n_{cag}(w - \varepsilon k^{\varepsilon - 1}) / 2 \quad (23)$$

Among them, the variable ω is a factor that affects the optimization coefficient, which is determined by factors such as the amount of data processed by the single-layer perceptron and the number of classifications. Let variable θ_2 be the correction error term,

$$w = \lambda(\sqrt{n_f l_{img}^2} - n_{cag}) - k^{\varepsilon} + \theta_2 \quad (24)$$

$$\lambda = k + \sum_{i=0}^{\varepsilon - 1} i \quad (25)$$

Let Opt_4 be the parameter matrix of the last fully connected layer, and the optimized fully connected layer parameter expression is:

$$Opt_4 = 2\sqrt{\frac{\rho}{n_{cag} + n_f l_{img}^2}} \times Opt_3 \quad (26)$$

D. Convolutional neural network training process

Convolutional neural network is essentially a mapping from input to output, which can learn many features that do not require any precise mathematical expression between input and output, and realize the mapping between input and output. Because the network performs supervised learning, its sample set is a vector pair of input vectors and ideal output vectors. The network training process is shown in Figure 10.

Use small random numbers of different sizes to initialize the connection weights of the convolutional layer threshold, two-layer convolution kernel, network input layer and hidden layer, and hidden layer and output layer in the network. At the same time, set the learning speed and the corresponding accuracy control parameters.

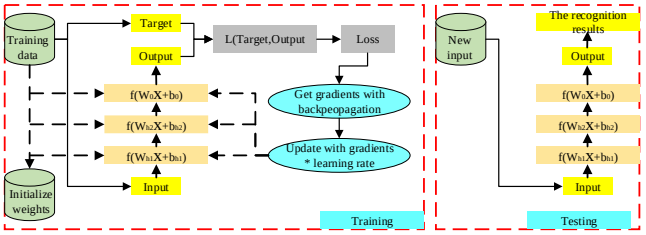

Figure 10. Network training flowchart

Because each convolutional layer has its threshold that can be trained, the weight of each convolution kernel is a learnable parameter. Therefore, the focus of CNN weight

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

update is the update of convolution kernel weight and convolution layer threshold.

(1) Update of weight and threshold of network

The reverse adjustment of neural network is the idea of gradient descent. For the network weight update in the algorithm, the adjustment of parameters always proceeds in the direction of error reduction.

(2) Weight update of convolution kernel

To calculate the weight update of the convolution kernel, the derived derivative of the sensitivity related to the error sensitivity of the convolutional layer, the output layer, and the down-sampling layer is used. For the convolution kernel, we mainly adjust the weight of the convolution kernel. If the dimension of each convolution kernel is n, then the training parameter it can learn is $n \times n$. The derivative of the weight between i-th feature map of layer l and the j-th feature map of $l+1$ layer is required. The calculation method is shown in equation (27).

$$\frac{\partial loss}{\partial k_{ij}} = x_i^l \square \delta_j^{l+1} \quad (27)$$

Among them, the symbol · refers to the convolution operation of the matrix. Suppose the size of i-th feature map x_i^l of layer l is 4×4, as shown in Figure 11. The j-th feature map δ_j^{l+1} of $l+1$ layer has an error sensitivity of 3×3, as shown in Figure 11.

Then the derivative size of the weight K_{ij} of the convolution kernel is 2×2, and the calculation method is shown in Figure 11. The variable δ_j^{l+1} is on the variable x_i^l. Pan the variable δ_j^{l+1} from left to right and from top to bottom. Moreover, corresponding to the value of the variable x_i^l multiplied, after accumulation to obtain the derivative of the convolution kernel function. After the derivative of the weight value of the convolution kernel is obtained, it is updated to the K_{ij} position corresponding to the original convolution kernel.

To update the bias of the threshold of the convolutional layer, simply add the error sensitivity of the j-th feature map of the above $l+1$ layer.

The threshold update derivative is calculated as shown in equation (28).

$$\frac{\partial loss}{\partial k_{ij}} = \sum_{u,v} (\delta_j^{l+1})_{u,v} \quad (28)$$

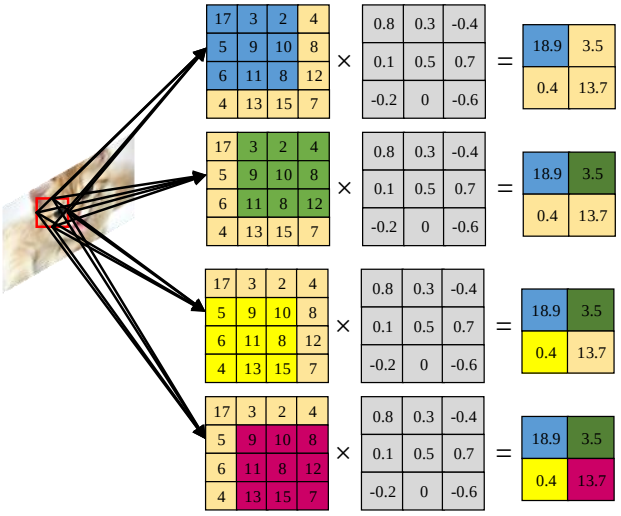

Figure 11. Calculation of feature map and weight derivative

IV. EXPERIMENTS AND RESULTS

A. Image data set and experimental environment

The experiments in this paper use the CIFAR-10 image dataset. CIFAR-10 is an image data set containing 60,000 color pictures. The size of each picture is 32×32. It is divided into 10 categories, and each category contains 6000 pictures.

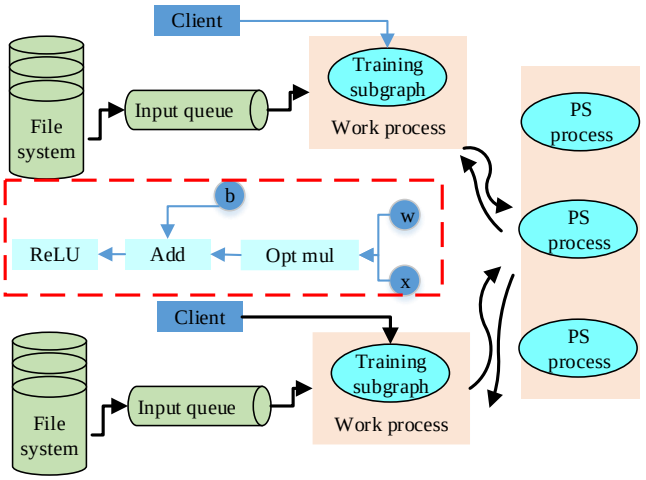

Figure 12. Data flow graph of TensorFlow

CIFAR-10 is divided into five training files and one test file. Each file contains 10,000 pictures. Among them, the test file is composed of 1000 pictures randomly selected from each category. The training file contains the remaining pictures and is out of order. Therefore, although each training file contains 5000 pictures, some files may have more pictures in a category than others.

The TensorFlow framework is used in this experiment. An open source software library uses data flow graphs for numerical calculations. A directed graph is used to represent the calculation of data flow. This graph consists

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

of a set of nodes. A data flow diagram describing the convolution operation is shown in Figure 12.

The hardware environment used in this experiment is as follows: Intel Socket 2011-v3 i7 processor, 128GB of memory, four NVIDIA Geforce TITAN X 12GB GPU and Ubuntu14.04 operating system.

The evaluation criteria of the experiment in this paper are the test accuracy and the size of the model finally generated by training.

B. Optimal selection of parameters

(1) Activation function

We use the sigmoid function, Tanh function and ReLU function as the activation function of the network respectively. The experimental results are shown in Table 2.

Table 2. Experimental results of different activation functions

Activation function type	Sigmoid function	Tanh function	ReLU function
False recognition rate	1.725%	1.658%	1.256%

It can be seen from Table 2 that the ReLU function has better function performance compared with other functions. The main reason is that the ReLU function forces certain function values to zero. Therefore, the network still has a good performance without using the regularization method, which can not only prevent the network from overfitting and accelerate the calculation speed, and the ReLU function can effectively prevent the gradient from disappearing. The sigmoid function and Tanh function will cause the problem of gradient disappearance when there are many network layers.

(2) Sampling method

After the feature map passes through the convolutional layer, the dimension is generally very large, which can easily cause the dimension disaster. Therefore, each convolutional layer in the convolutional neural network will be connected to a sampling layer to down sample the feature map to reduce the dimension of the feature map and reduce the computational complexity. Therefore, the sampling layer is also an essential part of the convolutional neural network structure. Choosing the appropriate sampling method will greatly improve the performance of the convolutional neural network. By sampling the feature map, the convolutional neural network can tolerate small deformations. Common sampling methods are maximum sampling, mean sampling, and random sampling. Choosing the most suitable sampling method can improve the recognition efficiency and accuracy of the network. In this paper, three different network models were constructed using three sampling methods. Except for the different sampling methods, the remaining parameters are all the same. The experimental results are shown in Table 3 below.

Table 3. Experimental results of different sampling methods

Sampling method	Mean sampling	Maximum sampling	Random sampling
False recognition rate	1.412%	1.376%	1.216%
Average time per iteration	38 s	76 s	101 s

Three experiments were carried out for each model, and the average value of the three experiments was selected as the final recognition result. It can be seen from the above table that the recognition effect of maximum sampling and random sampling is better than that of average sampling, and the recognition rate of random sampling is slightly higher than the maximum sampling. However, the computational complexity of random sampling is higher than the maximum sampling, the convergence speed is slower, and each iteration takes longer. For comprehensive consideration, the maximum sampling method is adopted in this paper.

(3) Pooling method and size selection

According to the structure of the network, three methods of pooling are selected in its pooling layer, mean pooling, max pooling, and stochastic pooling, and the size of its pooling is changed to 2×2, 3×3, 4×4, and 5×5, respectively.

Table 4. Classification results of several pooling methods on the CIFAR-10 database

	Train Error	Test Error
Mean pooling	6.521%	30.99239%
Max pooling	2.168%	28.961%
stochastic pooling	5.632%	28.135%

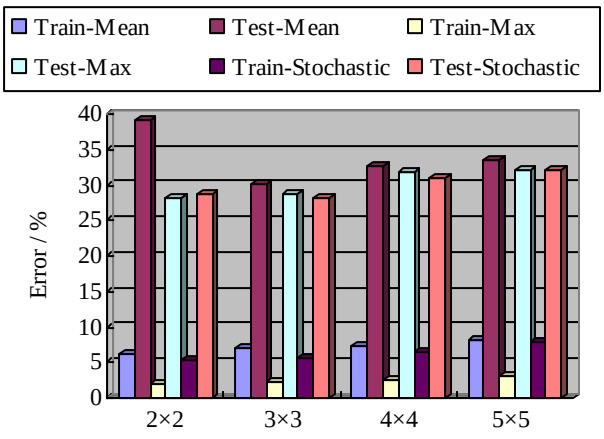

Figure 13. Classification results of different pooling sizes on the CIFAR-10 database

The best results of the tests on the CIFAR-10 database are shown in Table 4 below. The test results of different pooling sizes are shown in Figure 13. In the pooling method,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

choose the largest pooling and the average pooling. The smaller the pooling size is, the better the effect is. For the random pooling method, the optimal size is 3×3. A smaller pooling size will cause overfitting, and a larger pooling size will increase the error due to too much noise in down sampling.

C. Model convergence training

The training set samples are used for training, and the initial standard deviation of the initial weights of the network is 0.01, with a Gaussian distribution with an average value of zero. Suppose the number of sample iterations is 3000, the initial learning rate of the weight parameter is 0.001, and the momentum factor is 0.9. The training results are shown in Figure 14.

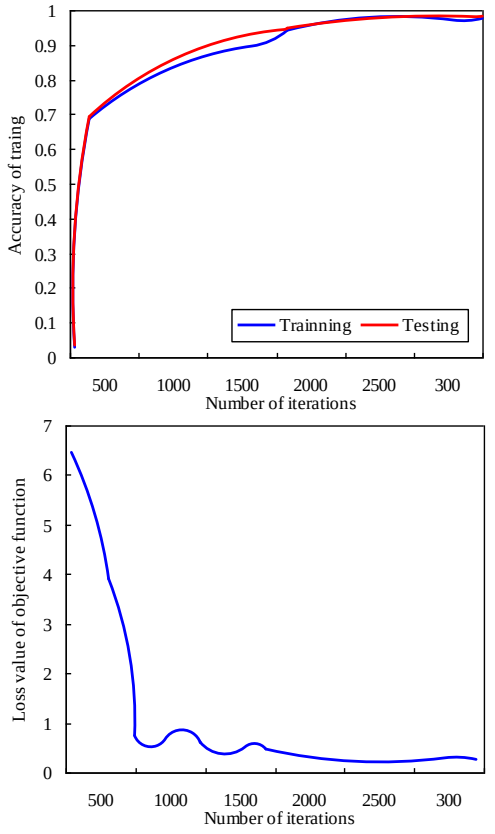

Figure 14. Preparation rate of network training and loss curve

From the simulation results in Figure 14, it can be seen that the training accuracy of the designed algorithm increases rapidly with the increase of the number of iterations, and tends to be stable. The classification results of the training set and the verification set are very close, with an accuracy rate of 0.985. The loss value of the objective function decreases rapidly. The loss value of the objective function converges to about 0.05 at the iteration of 2500 times.

D. Double-optimized performance analysis

According to different iteration times, three sets of iterative experiments were performed on convolutional optimization, convolutional optimization, and fully optimized convolutional neural networks.

The mean square error curve varies with the training batch at different iteration times. When the training times are once, twice and three times, the convergence curve of each algorithm is shown in Figure 15.

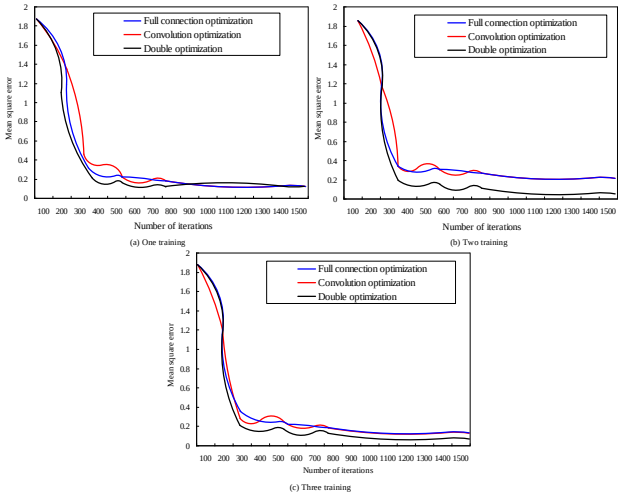

Figure 15. Convergence curve of each algorithm when training times are different

It can be seen from Figure 15 that in the iterative experiment of three trainings, the fully connected optimization algorithm has a higher mean square error at the initial stage. Nevertheless, the rate of decline is faster. The convolution optimization algorithm declines faster than the original algorithm and the fully connected optimization algorithm, and does not increase the mean square error at the initial stage like the fully connected optimization algorithm. Compared with the other two algorithms, the double optimization algorithm has a slightly faster convergence speed than the convolution optimization, but it is faster than the other two optimization algorithms and is the fastest algorithm.

E. Performance analysis of image recognition

Since we have given a variety of ResNet with different topologies, we analyzed them from different depths and different ShortCut. The network is $6n+2$ layers. When n takes different values, the experimental results are shown in Figure 16.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

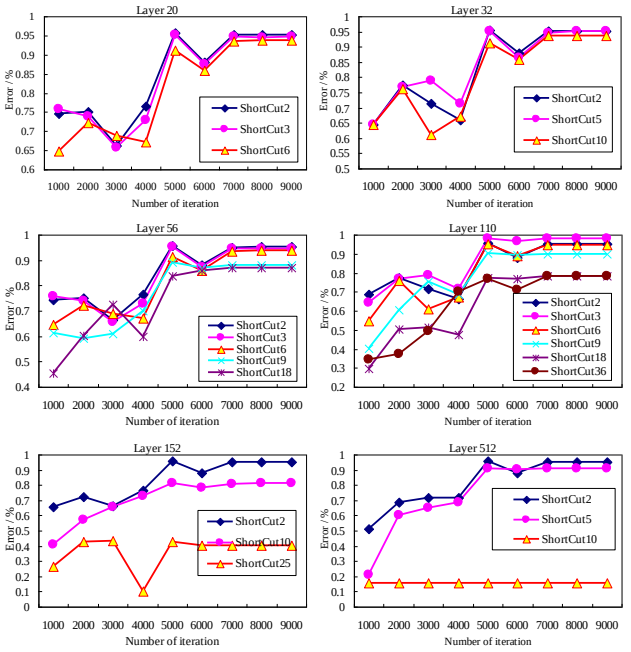

Figure 16. Accuracy of ResNet on CIFAR-10 under different deep networks and different ShortCut

While caring about recognition accuracy, we also care about the training time of ResNet with different topologies. The experimental results show that under the same depth of network, all structures of the network have the same training time. Figure 17 shows the training time of all networks under the 20-layer and 110-layer.

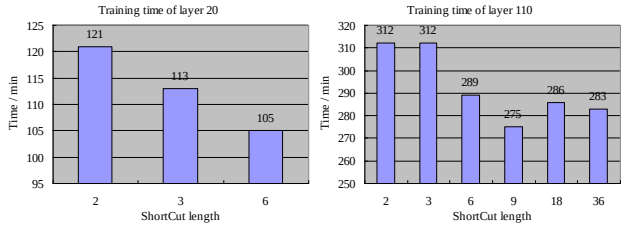

Figure 17. Comparison chart of training time

In ResNet, the recognition rate of images can be improved as the network deepens. For 56-layer and 110-layer networks, the length of ShortCut that can be obtained is basically the same. Therefore, we can see that under the smaller ShortCut, the original ResNet and the ShortCut3-ResNet constructed in this paper will have higher accuracy under deeper networks. Moreover, ShortCut3-ResNet improves more obviously. When ShortCut9 and ShortCut18 are selected, the accuracy of the 110-layer network is even lower than that of the 56-layer network. For 32-layer and 152-layer networks, the values are the same. However, when ShortCut10 is taken, the recognition rate of the 32-layer network is better than that of the 152 layer, as shown in Figure 18.

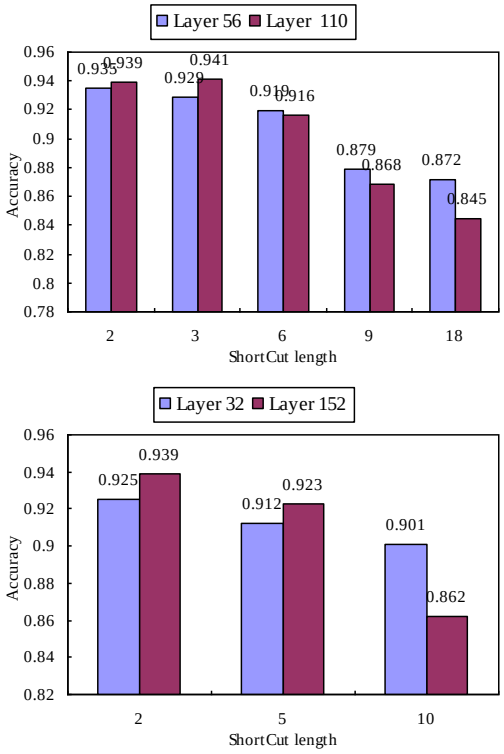

Figure 18. Experimental analysis of the same network under different ShortCut

From the analysis in Figure 19, it can be seen that the original ResNet and ShortCut3-ResNet are getting better and better as the network slowly gets deeper. Nevertheless, for the residuals and networks of other topologies, as the network deepens, the recognition effect will become worse with the increase of ShortCut, indicating that the size of ShortCut has a great influence on the recognition results. When the length of ShortCut is less than six, the recognition accuracy rate is basically improved with the deepening of the network. When it is equal to six, it remains unchanged, and when it is greater than six are, it decreases. The recognition accuracy of the 110-layer ShortCut3-ResNet constructed in this paper is equivalent to the accuracy of the original 152-layer ResNet. Nevertheless, the parameters of the 110-layer network are much less than the 152-layer network. Therefore, the 110-layer ShortCut3-ResNet designed in this paper is a good network.

In Table 5, the test results of the algorithm and other methods designed in this paper are given in the data set. The main evaluation criteria are test accuracy and the size of the model generated by the training.

In Table 5, the test methods are AlexNet, VGGNet, ResNet, and Random Forest. There are also algorithms for HSC proposed in reference [30], Scatt-Net proposed in reference [31] and PCAnet proposed in reference [32]. It can be obtained from Table 5 that the random forest is inferior to the traditional CNN in network performance, and the algorithm proposed in this paper is higher than the single structure algorithm in network test accuracy. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3006097, IEEE Access

AlexNet network uses a three-layer fully connected layer, so the network has a large number of training parameters, resulting in more storage resources occupied by the final training model. In a comprehensive comparison, the algorithm proposed in this paper is more diversified when extracting features and the test accuracy of the network has been improved. After combining the ultra-lightweight network structure, the amount of parameters is appropriately reduced.

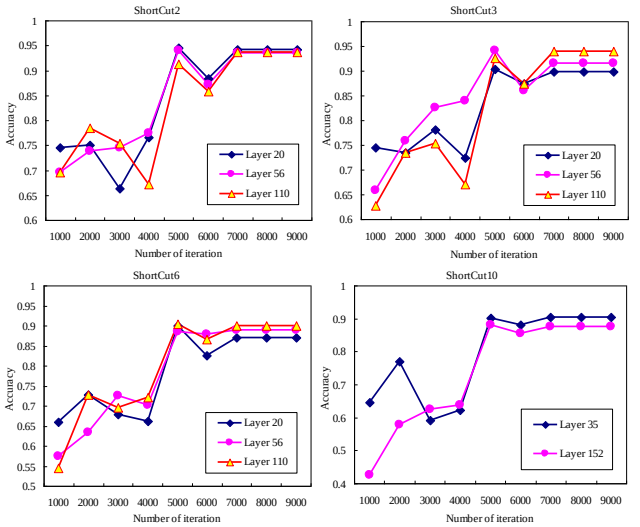

Figure 19. Experimental analysis of the same network under different deep networks

Table 5. Performance comparison of different algorithms on the CIFAR-10 dataset

Methods	Accuracy	Model size
Random forest	0.508	-
AlexNet	0.658	0.826 M
VGGNet	0.896	553.612 M
ResNet	0.926	95.625 M
HSC	0.932	239.012 M
Scatter-Net	0.975	85.621 M
PCANet	0.919	54.512 M
This paper	0.985	18.201 M

V. CONCLUSION

In order to improve the ability of the convolutional neural network to classify and recognize two-dimensional images and speed up the convergence of the algorithm, this paper proposes a new convolutional network algorithm. First, a recurrent neural network is introduced into the convolutional neural network, and the deep features of the image are learned in parallel using the convolutional neural network and the recurrent neural network. Not only can we use convolutional neural networks to learn high-level features, but also recursive neural networks to learn the combined features of low-level features. Secondly, according to ResNet's idea of skipping convolutional layers, we construct a new residual module ShortCut3-ResNet. Finally, the convolutional layer and the full connection process are optimized. Experiments show that the proposed convolutional neural network algorithm can improve the feature extraction accuracy and image recognition ability of convolutional neural network.

REFERENCES

[1] Pan J. How Chinese officials use the Internet to construct their public image[J]. Political Science Research and Methods, 2019, 7(2): 197-213.

[2] Liansheng S, Xiao Z, Chongtian H, et al. Silhouette-free interference-based multiple-image encryption using cascaded fractional Fourier transforms[J]. Optics and Lasers in Engineering, 2019, 113: 29-37.

[3] Zhu X, Li Z, Zhang X Y, et al. Deep convolutional representations and kernel extreme learning machines for image classification[J]. Multimedia Tools and Applications, 2019, 78(20): 29271-29290.

[4] Wang F, Jiang D, Wen H, et al. Adaboost-based security level classification of mobile intelligent terminals[J]. The Journal of Supercomputing, 2019, 75(11): 7460-7478.

[5] Wen L, Zhou K, Yang S. A shape-based clustering method for pattern recognition of residential electricity consumption[J]. Journal of cleaner production, 2019, 212: 475-488.

[6] Zan T, Liu Z, Wang H, et al. Control chart pattern recognition using the convolutional neural network[J]. Journal of Intelligent Manufacturing, 2020, 31(3): 703-716.

[7] Yu J, Zheng X, Wang S. A deep autoencoder feature learning method for process pattern recognition[J]. Journal of Process Control, 2019, 79: 1-15.

[8] Freire-Obregón D, Narducci F, Barra S, et al. Deep learning for source camera identification on mobile devices[J]. Pattern Recognition Letters, 2019, 126: 86-91.

[9] Zhe X, Chen S, Yan H. Directional statistics-based deep metric learning for image classification and retrieval[J]. Pattern Recognition, 2019, 93: 113-123.

[10] O'Hagan A, Murphy T B, Scrucca L, et al. Investigation of parameter uncertainty in clustering using a Gaussian mixture model via jackknife, bootstrap and weighted likelihood bootstrap[J]. Computational Statistics, 2019, 34(4): 1779-1813.

[11] Wang S, Gittens A, Mahoney M W. Scalable kernel K-means clustering with Nyström approximation: relative-error bounds[J]. The Journal of Machine Learning Research, 2019, 20(1): 431-479.

[12] Karimi F, Sultana S, Babakan A S, et al. An enhanced support vector machine model for urban expansion prediction[J]. Computers, Environment and Urban Systems, 2019, 75: 61-75.

[13] Sur P, Candès E J. A modern maximum-likelihood theory for high-dimensional logistic regression[J]. Proceedings of the National Academy of Sciences, 2019, 116(29): 14516-14525.

[14] Zhu D, Zhang F, Wang S, et al. Understanding place characteristics in geographic contexts through graph convolutional neural networks[J]. Annals of the American Association of Geographers, 2020, 110(2): 408-420.

[15] Jati A, Georgiou P. Neural Predictive Coding Using Convolutional Neural Networks Toward Unsupervised Learning of Speaker Characteristics[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019, 27(10): 1577-1589.

[16] Morvan M, Arangalage D, Franck G, et al. Relationship of iron deposition to calcium deposition in human aortic valve leaflets[J]. Journal of the American College of Cardiology, 2019, 73(9): 1043-1054.

[17] Hansen M F, Smith M L, Smith L N, et al. Towards on-farm pig face recognition using convolutional neural networks[J]. Computers in Industry, 2018, 98: 145-152.

[18] Parmar K, Kher H, Gandhi M. Facial Expression Recognition Using Convolutional Neural Network[J]. Journal of Open Source Developments, 2019, 6(1): 18-27.

[19] Grossberg S. The resonant brain: How attentive conscious seeing regulates action sequences that interact with attentive cognitive

learning, recognition, and prediction[J]. Attention, Perception, & Psychophysics, 2019, 81(7): 2237-2264.

[20] Nguyen H T, Nguyen C T, Ino T, et al. Text-independent writer identification using convolutional neural network[J]. Pattern Recognition Letters, 2019, 121: 104-112.

[21] Wang R, Xu J, Han T X. Object instance detection with pruned Alexnet and extended training data[J]. Signal Processing: Image Communication, 2019, 70: 145-156.

[22] Matlani P, Shrivastava M. Hybrid deep VGG-NET convolutional classifier for video smoke detection[J]. Computer Modeling in Engineering & Sciences, 2019, 119(3): 427-458.

[23] Khan R U, Zhang X, Kumar R. Analysis of ResNet and GoogleNet models for malware detection[J]. Journal of Computer Virology and Hacking Techniques, 2019, 15(1): 29-37.

[24] McNeely-White D, Beveridge J R, Draper B A. Inception and ResNet features are (almost) equivalent[J]. Cognitive Systems Research, 2020, 59: 312-318.

[25] Scardapane S, Van Vaerenbergh S, Totaro S, et al. Kafnets: Kernel-based non-parametric activation functions for neural networks[J]. Neural Networks, 2019, 110: 19-32.

[26] Montanelli H, Du Q. New error bounds for deep ReLU networks using sparse grids[J]. SIAM Journal on Mathematics of Data Science, 2019, 1(1): 78-92.

[27] Wang S H, Muhammad K, Hong J, et al. Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization[J]. Neural Computing and Applications, 2020, 32(3): 665-680.

[28] Côté-Allard U, Fall C L, Drouin A, et al. Deep learning for electromyographic hand gesture signal classification using transfer learning[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2019, 27(4): 760-771.

[29] Feng S, Zhou H, Dong H. Using deep neural network with small dataset to predict material defects[J]. Materials & Design, 2019, 162: 300-310.

[30] Zhang Y, Qu Y, Li C, et al. Ontology-driven hierarchical sparse coding for large-scale image classification[J]. Neurocomputing, 2019, 360: 209-219.

[31] Carroll E L, Gallego R, Sewell M A, et al. Multi-locus DNA metabarcoding of zooplankton communities and scat reveal trophic interactions of a generalist predator[J]. Scientific reports, 2019, 9(1): 1-14.

[32] Hu F, Zhou M, Yan P, et al. PCANet: A common solution for laser-induced fluorescence spectral classification[J]. IEEE Access, 2019, 7: 107129-107141.

[33] Wang Y, Wang G, Chen C, et al. Multi-scale dilated convolution of convolutional neural network for image denoising[J]. Multimedia Tools and Applications, 2019, 78(14): 19945-19960.

[34] Sadr H, Pedram M M, Teshnehlab M. A Robust Sentiment Analysis Method Based on Sequential Combination of Convolutional and Recursive Neural Networks[J]. Neural Processing Letters, 2019, 50(3): 2745-2761.

[35] Guo Z, Lv X, Yu L, et al. Identification of hepatitis B using Raman spectroscopy combined with gated recurrent unit and multiscale fusion convolutional neural network[J]. Spectroscopy Letters, 2020, 53(4): 277-288.

[36] Bernal J, Kushibar K, Asfaw D S, et al. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review[J]. Artificial intelligence in medicine, 2019, 95: 64-81.

[37] Kamilaris A, Prenafeta-Boldú F X. A review of the use of convolutional neural networks in agriculture[J]. The Journal of Agricultural Science, 2018, 156(3): 312-322.

[38] Samadi F, Akbarizadeh G, Kaabi H. Change detection in SAR images using deep belief network: a new training approach based on morphological images[J]. IET Image Processing, 2019, 13(12): 2255-2264.

[39] Jing W, Jiang T, Zhang X, et al. The optimisation of speech recognition based on convolutional neural network[J]. International Journal of High Performance Computing and Networking, 2019, 13(2): 222-231.

[40] Bacchi S, Oakden-Rayner L, Zerner T, et al. Deep learning natural language processing successfully predicts the cerebrovascular cause of transient ischemic attack-like presentations[J]. Stroke, 2019, 50(3): 758-760.

[41] Zhang Y, Zhang X, Sun H, et al. Portable brain-computer interface based on novel convolutional neural network[J]. Computers in biology and medicine, 2019, 107: 248-256.

[42] Xu C, Yang J, Lai H, et al. UP-CNN: Un-pooling augmented convolutional neural network[J]. Pattern Recognition Letters, 2019, 119: 34-40.

[43] Zhang Y D, Dong Z, Chen X, et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation[J]. Multimedia Tools and Applications, 2019, 78(3): 3613-3632.

[44] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. ACM Transactions on Graphics (TOG), 2019, 38(5): 1-12.

[45] Basha S H S, Dubey S R, Pulabaigari V, et al. Impact of fully connected layers on performance of convolutional neural networks for image classification[J]. Neurocomputing, 2020, 378: 112-119.

[46] Su L, Ma L, Qin N, et al. Fault diagnosis of high-speed train bogie by residual-squeeze net[J]. IEEE Transactions on Industrial Informatics, 2019, 15(7): 3856-3863.

Youhui Tian,received the master's degree from the Heilongjiang University of Science and Technology in 2013. He is currently a Senior Engineer with the Jiangsu Vocational Institute of Commerce. His research area is network technology and information systems.