Academic Journal of Computing & Information Science
ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

# An image recognition method using parallel deep CNN

**Zhiqiang Gao[1,2], Yuexin Li[2], Shijie Wang [1]**

*1 China University of Petroleum, Beijing, 102249, China*
*2 Hubei University, Hubei, Wuhan, 430062, China*

***ABSTRACT.*** *Based on the characteristics of the image, the traditional methods are improved based on the theory of convolutional networks and the research results at home and abroad. Firstly, the training algorithm of convolutional network is studied. By analysing the algorithm, debugging and finding the optimal initialization parameters and the most suitable network structure configuration. Then, for the calculation of the classification result, a multi-region test method is used, and the accuracy of image recognition can be improved by calculating a plurality of regions of the image during the test. Finally, a general dataset input interface is designed for the system, and the experimental results show that the improved convolutional neural network structure is more conducive to obtaining the diversity characteristics of the image. Under the same experimental conditions, the recognition rate is higher than that of the traditional network.*

***KEYWORDS:*** *Image recognition; Deep convolutional neural network; Feature fusion; Diversity feature*

## 1. Introduction

The human brain receives a lot of information all the time and can process and save it. This information can be read and used when it is used next time. Artificial neural network is an artificial intelligence method that imitates the structure and unit of the human brain [1,2]. Since Rumelhart and his colleagues proposed learning algorithms in 1985, research on neural networks has raged, and artificial neural networks have gradually been applied to various fields. However, using the traditional artificial neural network, in the process of image processing, despite a lot of pre-processing work, there are still many noise factors in the image that affect the image recognition effect [3,4]. In order to further improve the efficiency of image recognition, many scholars at home and abroad have proposed many different feature extraction algorithms and neural network models.

Among them, the literature [5] proposed a space-time domain deep convolutional neural network based on the filter response product. The network first divides the convolution kernels corresponding to adjacent frames into two groups, and detects the motion characteristics between adjacent frames by the rotation angle of the adjacent frames on the invariant subspace. Literature [6] proposes a method for learning binary hash coding based on deep convolution path network for large-scale image retrieval. The basic idea of this paper is to add a hash layer to the deep learning framework while learning image features and hash functions. In [7], the advantages of deep learning convolutional neural network automatic learning feature are used to solve the problem of poor universality of manual design features, and the DNN parallel computing strategy based on CUDA architecture is used to improve training speed and speed up classification, and for deep convolution. Neural networks are susceptible to parasitic disturbances and introduce batch regularization to improve the robustness of the algorithm [8-10].

Based on the reality, this paper improves the existing image recognition process combined with artificial neural network. Due to the characteristics of artificial neural networks, the network can automatically learn various low-level and high-level features. Therefore, image preprocessing can be merged into the image feature extraction part, and the image feature extraction part can be realized by artificial neural network model. Due to the complexity and redundancy of the artificial neural network, on the one hand, the model is too large, and when it is applied on the mobile terminal, due to the limitations of its own platform, the space and memory of the network computing time are difficult to meet the calculation requirements, and the other Due to the speed of the network, it is generally difficult to meet the real-time requirements. Therefore, this paper proposes a new system flow that adds the network model compression and acceleration module to the image recognition flexibly, and adopts the model compression and acceleration method for the artificial neural network. Its applicability and real-time requirements. Through the actual model test, it is verified that the proposed model is feasible and effective.

## 2. Convolutional network layer design

### *2.1 Input layer and output layer design*

The network model for image classification in this paper includes three convolutional layers. There is a collection layer behind each convolutional layer. The convolutional layer uses a filter size of 5*5. All collection layers have a stride size of 2 and a collection area of 3*3. The first convolutional layer is followed by the largest collection, and the others are averaged. The reaction normalized layer is applied to the first two collection layers. The local weight non-sharing process is used in the convolutional layer. The third collection layer is followed by a fully connected layer fc and an enhancement layer. We mainly use DROPOUT in the fc layer. There is also a logistic regression cost layer, which is mainly used to perform multi-regional logistic regression calculation of images in this layer.

The output of the convolutional neural network uses the “1 of N” method, and the output is a one-dimensional matrix vector containing 10 output values (the value range of each value is [0-1]). We define two fully connected layers in the classification: the fc10 and probs layers, where the probs layer is a data

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

enhancement layer. The Fc10 layer is connected to the third-stage sub-sampling layer, no matter how many outputs are in the S3 layer, that is, no matter how many inputs are received by the fc10 layer, we only define the number of outputs by the parameter's outputs. Probs is an enhancement layer. Such a layer is very effective for data classification. Its input is the fc10 layer, and the data generated by the fc10 layer appears as a probability. There is also a logical regression cost layer behind the probs layer. If the probability value of the output of the i-th position is the largest (i starts from 0), it means that the set of feature vectors belongs to the i-th classification. In the system of this paper, there are 10 categories of 0-9, each of which corresponds to the label of the position corresponding to the vector matrix in the label layer.

Because the system finally needs to output the classification error rate recognized by the system, we need to compare the actual label of the image with the label output by the system. After the input layer we also need to define a logical cost regression layer. At this level we need to enter two layers: the label layer and the enhancement layer. The label layer is the vector matrix of the data set that is the label, and the enhancement layer is the output of the system test generated after the image is trained. Comparing the two data to get the classification and recognition of the image is correct, and then the system generates a classification error rate [11].

The input layer and output layer of the neural network are determined by the characteristics of the network structure and the network samples, and the weight connection between the neural units is learned by training, that is, the memory function of the network. The specific problem directly determines the dimensions of the input and output layers of the neural network. It can have one or several hidden layers. The number of hidden layers and the number of neurons contained in the hidden layer are related to the problem to be classified. So far, there is no general method for determining the functional relationship between them [12]. Basically, it is adjusted in experiments, and sometimes the network needs to be reduced. The recognition accuracy and generalization ability are not the deeper the network layer, the better, and the number of hidden layers and the number of neurons in each layer are better.

Sometimes, the generalization ability and recognition accuracy of the network need to be reduced. The MINIST data set is a 28×28 image, which is stored in the data file in the pixel manner of the entire image, and the pixel is a decimal number between 0 and 255, and is stored in the file in hexadecimal form. Then each picture corresponds to 28×28, which is 784 pixels, taking all the pixels of a picture as input, and the output of the network is between 0-9, that is, the network output has 10 digital categories, so the output layer needs to set 10 neuron nodes.

### *2.2 Intermediate layer design*

2.2.1. The choice of activation function:

After determining the network topology and learning algorithm, choose an appropriate activation function to determine the convergence and convergence speed of the network. If an inappropriate activation function is selected, it may cause over-fitting of the network and even cause the network to lose learning ability. As for the choice of activation function, there is no certain applicable function for different identification problems. Only in the experiment, it is explored according to the specific problem.

Each neuron can receive one or several input signals from other neurons in the network. There is a connection weight between the neuron and the neuron, and the neuron is in an active or suppressed state. The weighted sum of the inputs is determined. The output value of the perceptron network can only be 0 or 1, and it is not transmissive. However, the sensitivity-based training algorithm requires that its output function must be everywhere, so the S-type derivative function is introduced, that is, Each neuron must be processed by the S-type activation function before it can be output.

Commonly used activation functions are tanh function and sigmoid function. Compared with other activation functions, sigmoid has its own advantages, such as robustness, smoothness, and its derivatives can be represented by itself. This feature is important during the experimental training phase because the back propagation of weights produces many derivatives when seeking the derivative of the activation function. A typical continuous function creates many computational and storage problems. In addition, selecting the sigmoid function will speed up the convergence under the same conditions. After the sigmoid function, the output of the neuron is as shown in equation (1).

$$y = \frac{1}{1+e^{-kx}} \tag{1}$$

$x$ is the net input of neurons. The larger the value of $k$ is, the flatter the function is, the easier the network is to converge, but the convergence speed is slow; the smaller the value of k, the steeper the function, and the more difficult the network is to converge, but The convergence speed is faster. Experiments show that when the error is small when $k=1$ and the convergence speed is faster, the activation function finally selected in this paper is shown in equation (2).

$$y' = \frac{1}{1+e^{-x}} - \frac{1}{\left(1+e^{-x}\right)^2} = y - y^2 \tag{3}$$

The activation derivative can be expressed by itself. This feature greatly reduces the derivative operation of the activation function in the error propagation process, which saves the computer a lot of time and greatly speeds up the convergence of the network.

2.2.2. The design of the convolution layer:

The convolution layer convolves the feature map by the convolution kernel, and then processes the convolution result with the sigmoid function, and finally forms a feature map. Compared with the original image, the feature of the convolved feature

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

map becomes more and more significant, just like the learnable convolution kernel extracts the feature information of the sample. This is also the advantage of CNN, which is to learn the sample features while calculating, and does not use explicit extraction features. The convolution kernel is an n×n matrix, and n is usually an odd number, and the values therein can be continuously updated during the learning process[13,14].. During the training iteration process, the sample characteristics are continuously enhanced to approximate the accurate output.

Here, discrete convolution is used, and the step size of the convolution is l, that is, one pixel is moved each time in the horizontal and vertical directions. If the original image size is 3×3 and the convolution kernel size is 2×2, then the volume The result of the product operation is shown in Figure 1.

$$\begin{bmatrix} 2 & 0 & 3 \\ 1 & 4 & 2 \\ 5 & 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2\times1+0\times2+1\times3+4\times4 & 0\times1+3\times2+4\times3+2\times4 \\ 1\times1+4\times2+5\times3+6\times4 & 4\times1+2\times2+6\times3+2\times4 \end{bmatrix}$$

*Figure 1 Convolution operation*

2.2.3. The design of the down sampling layer:

Using the principle of local correlation of images, the image is subsampled after convolution, which not only preserves the effective information of the image, but also reduces the amount of data processing and the learning dimension of the network. Mean-pooling selects four pixels of the convolutional layer image and takes the average value as one pixel of the down sampling layer, which greatly reduces the network scale. If the windowing window size is 2×2, in the forward direction, the 2×2 window average is not coincident on the previously convolved output, and the obtained value is the value after the current mean-pooling. In backwards, divide this value into four equal parts and place it in the front 2×2 grid. The 2×2 window average is not coincident in order, and the obtained value is the value after the current mean-pooling. In the backwa, divide this value into four equal parts and put it in the front 2×2 grid.

$$\text{forward:}\left[1\ 3;\ 2\ 2\right]\text{->}\left[2\right]$$

$$\text{backward:}\left[2\right]\text{->}\left[0.5\ 0.5;\ 0.5\ 0.5\right]$$

In this paper, max-pooling is chosen, which is to maximize the feature points in the neighbourhood. In the forward, you only need to take the largest one in the 2×2 window. In the backward, you should put the current value to the largest position before, and the other three positions are 0.

$$\text{forward:}\left[1\ 3;\ 2\ 2\right]\text{->}3$$

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

$$\text{backward}:\left[3\right]\text{->}\left[0\ 3;\ 0\ 0\right]$$

### *2.3 Image pre-processing*

This paper is based on the CNN of the literature [15], which does not give an input interface for a common data set. In this section, a general method for constructing data sets is proposed. We can construct data sets and input them into the system for image recognition and classification. The following is a detailed process of building a data set by building a small data set. This method can also be used for other data sets. The data set is built as shown in the following figure:

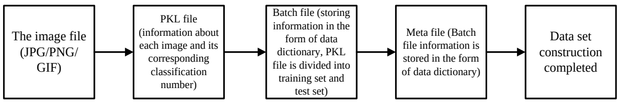


*Figure 2 The process of building data sets*

Pre-processing is an essential step in image recognition, which requires converting the original image into a binary form that the recognizer can receive. Image pre-processing is to remove redundant information and noise in digital images, obtain a normalized lattice, and prepare for image recognition. Therefore, in the pre-processing process, when eliminating factors unrelated to identification, the digital features of the original image should be preserved as much as possible. In order to ensure the diversity of handwritten numbers, this paper selects the MNIST handwritten digit set (a subset of the NIST database), which has 60,000 training sample sets and 10,000 test sample sets [16,17].

The pre-processing operations include grayscale of the image, tilt correction, normalization, binarization, refinement, etc. For the handwritten digit set, the effect of the grayscale processing is not obvious, and the image is For the same size, no normalization operation is required. Therefore, the main purpose of this paper is the image refinement and tilt correction operation.

## 3. Convolutional neural network training process

The convolutional neural network is essentially a mapping from input to output. It 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, so its sample set is a vector pair like an input vector and an ideal output vector.

### *3.1 Network Initialization*

The small random number with different sizes is used to initialize the convolution layer threshold, the two-layer convolution kernel, the BP network input

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

layer and the hidden layer, and the connection weight between the hidden layer and the output layer. At the same time, the learning speed and the corresponding precision control parameters are set.

### *3.2 Network Forward Propagation Process*

First, a 6-layer convolutional neural network (without the input layer) is initialized, and then the network is trained. The training begins with the forward propagation phase. The pixel matrix with the sample size of 28×28 is regarded as the input layer of CNN, and the original image is convolved by 6 convolution kernels with a size of 5×5, so that six convolutional layers C1 with different characteristics are obtained, and the size is 24 ×24, and these feature maps store features extracted by different filters.

Next, a 2×2 local averaging is started for each feature map in C1, thereby obtaining six 12×12 down sampling layers S2, which are 1/4 of the C1 layer in dimension. Then, a second convolution operation is performed on S2, and 12 5×5 convolution kernels are selected to obtain 12 feature maps C3 having a size of 8×8. A 2×2 down sampling process is also performed on the C3 layer to obtain S4 having a size of 4×4. Then, all the 4×4 feature maps are broken into a one-dimensional matrix E5, which is regarded as the input layer of the BP neural network. Finally, the output layer F6 with 10 neurons as the representative category is used, and E5 and F6 are all the method of connection.

The forward propagation phase of convolutional neural networks is like the forward-to-back transmission of traditional neural networks. The samples in the data set are input into the network, and the output is obtained through step-by-step transformation. The output of the upper layer in the network is used as the output. The input of the current layer, the relationship between input $x^{l-1}$ and output $x^l$ of the current layer is as follows:

$$x^l = f\left(W^l x^{l-1} + b^l\right) \tag{4}$$

Where $l$ is the number of layers; $W$ is the weight; $b$ is an offset; $f$ is an activation function.

### *3.3 Reverse Propagation Phase*

In the forward propagation process, usually the convolution layer is followed by a corresponding down sampling layer. Since the convolution operation can extract the features of the original image, the down sampling layer performs spatial scaling, which can make the network structure more Robust. In the process of backpropagation, different calculation methods are used to adjust the coefficients of the convolutional layer and the down sampling layer. We call the magnitude of the weight adjustment as the sensitivity, that is, modify the network weight of the corresponding layer, which is calculated according to each the error sensitivity of a

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

layer.

In the convolutional layer, the feature map of this layer can be obtained by convolution operation of several feature maps of the upper layer, and each output feature map will be given an offset variable b. For the feature map, even if the same is used for the feature map Input feature map, and because it uses different convolution kernels, it will also form different output feature maps, which also determines the characteristics of different dimensions extracted from each output feature map compared to the input feature map. It ensures that the key features of the original image are not lost during the computational transmission between networks. Because the convolution layer is the down sampling layer, the error is transmitted through the down sampling layer. The error propagation is the reverse process of down sampling, that is, up sampling, that is, the error of the down sampling layer is copied into 2×2=4 copies, making it the same size as the convolutional layer. The convolutional layer output is generated by the action of the sigmoid function. According to the error back propagation algorithm, the error from the down sampling layer needs to be processed by the sigmoid.

After each forward propagation, an error needs to be defined to characterize the state of the network after this propagation. The backpropagation process is to pass the error forward by layer by means of reverse transmission, so that the neurons in the upper layer update their own weights according to the error. The back-propagation algorithm of convolutional neural networks generally adopts a method based on gradient descent. By calculating the global error of the network, the unit weight in the network can be adjusted in the direction of error reduction. The iteration is divided into two steps: finding the error between the output obtained by the forward propagation phase of the sample and the target output, and transmitting the inverse of the error to realize the update of the weight in the network.

(1) For sample $n$, the error function can be expressed as:

$$E^n = \frac{1}{2}\sum_{k=1}^{c}\left(t_k^n - y_k^n\right)^2 = \frac{1}{2}\left\|t^n - y^n\right\|_2^2 \tag{5}$$

The number of sample categories in the data set is c, $t_k^n$ is the expected value corresponding to the $k$ dimension of the $n$ sample in the data set, and $y_k^n$ is the $k$ dimension of the actual output corresponding to the $n$ sample in the data set. The output of the current layer can be Expressed as:

$$x^l = f\left(u^l\right),\ u^l = W^l x^{l-1} + b^l \tag{6}$$

Here, $f$ is the activation function, and then calculate the partial derivative of the error with respect to the network bias and weight.

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial b} = \delta \tag{7}$$

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

Since $\frac{\partial u}{\partial b}=1$, $\frac{\partial E}{\partial b}=\frac{\partial E}{\partial u}\frac{\partial u}{\partial b}=\delta$, then the sensitivity of the error signal and the partial derivative of the error with respect to all input of A single neuron are equal, then the sensitivity formula of the hidden layer can be deduced as follows:

$$\delta^l=\left(w^{l+1}\right)^T\delta^{l+1}\circ f'\left(u^l\right) \quad (8)$$

Here, "$\circ$" means that each element is multiplied. Since the sensitivity of the network is generally determined by this layer, the sensitivity of the output layer is further expressed as:

$$\delta^l=f'\left(u^l\right)\circ\left(y^n-t^n\right) \quad (9)$$

Since there is no error function that can be directly used in the hidden layer, the back-propagation algorithm needs the above formulas to realize the back-propagation layer by layer. Finally, the weight of each neuron in the hidden layer is updated. For the l layer of the network, the partial derivative of the error relative to each weight in the layer is equal to the cross product of the input and sensitivity of the layer, and then the calculated partial derivative is multiplied by a negative learning rate to obtain the updated weight of the neuron in the layer.

$$\frac{\partial E}{\partial w^l}=x^{l-1}\left(\delta^l\right)^T \quad (10)$$

$$\Delta w^l=-\eta\frac{\partial E}{\partial w^l} \quad (11)$$

Because the convolutional neural network is composed of convolution and sample layer alternate connection, the characteristics of the sample layer on regional characteristics of the figure is based on convolution layer and sampling operation, every nerve in the sample layer node corresponds to the convolution layer in a local area, which makes the convolution of feature points and sampling layer exists between the feature points many-to-one mapping relation. This requires that the errors of each nerve node in the lower sampling layer be formed into an error signal graph corresponding to the local region in the convolution layer during the backpropagation. For example, when calculating the error of the layer $l$, the error of each node in the lower sampling layer shall be firstly sampled to obtain the error signal graph corresponding to the size of the corresponding region in the convolution layer. Then the partial derivative of the activation function in the convolution operation is multiplied by the resulting error signal graph. Since the weights in the lower sampling layer are all $\beta$, multiply the above calculation results by the previous $\beta$:

$$\delta_j^l=\beta_j^{l+1}\left(f'\left(u^l\right)\circ up\left(\delta_j^{l+1}\right)\right) \quad (12)$$

When a characteristic graph is given, the deviation gradient relative to the bias

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

can be calculated, that is, the sum:

$$\frac{\partial E}{\partial b_j}=\sum_{u,v}\left(\delta_j^l\right)_{uv} \tag{13}$$

Finally, back-propagation algorithm is used to calculate the weight gradient. Considering the weight sharing in each feature graph, it is necessary to sum up all gradients related to the weight:

$$\frac{\partial E}{\partial k_{ij}^l}=\sum_{u,v}\left(\delta_j^l\right)_{u,v}\left(p_i^{l-1}\right)_{uv} \tag{14}$$

Where $p_i^{l-1}$ is the region where $x_i^{l-1}$ multiplies with $k_{ij}^l$ element by element in the convolution operation, and the value of $(u,v)$ position in the output convolution feature graph is obtained by multiplying the region of $(u,v)$ the previous layer by the convolution kernel $k_{ij}^l$. For the above formula, MATLAB convolution function can be used to achieve:

$$\frac{\partial E}{\partial k_{ij}^l}=rot180\left(conv2\left(x_i^{l-1},rot180\left(\delta_j^l\right),'valid'\right)\right) \tag{15}$$

When calculating the partial derivative of the lower sampling layer, since the convolution layer is connected behind the lower sampling layer, it is necessary to first calculate which local region of the lower sampling layer corresponds to which pixel in the error of the next layer.

$$\delta_j^l=f'\left(u_j^l\right)\circ conv2\left(\delta_i^{l+1},rot180\left(k_j^{l+1}\right),'full'\right) \tag{16}$$

Then figure out the gradient corresponding to the multiplicative bias $\beta$ and the additive bias $b$, and the gradient of the additive bias $b$ is equal to the sum of the elements in the error signal diagram:

$$\frac{\partial E}{\partial b_j}=\sum_{u,v}\left(\delta_j^l\right)_{uv} \tag{17}$$

Since the multiplicative bias A has $\beta$ certain correlation with the original down-sampling graph without additive bias in the current layer, these feature graphs can be directly saved in the forward propagation stage, and the calculation of these feature graphs is omitted:

$$d_j^l=down\left(x_j^{l-1}\right) \tag{18}$$

Therefore, the gradient of the error pair can be expressed as:

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

$$\frac{\partial E}{\partial b_j} = \sum_{u,v} \left( \delta_j^l \circ d_j^l \right)_{uv} \tag{19}$$

In the process of network training, the initial network weight is generally set to a random value, but the actual output result of the network will be greatly different from the expected output result. The proposed error back-propagation algorithm provides the possibility for the training and learning of neural networks [18]. Because of its excellent performance, it has become the most extensive training method in artificial neural networks.

## 4. Simulation results and analysis

### *4.1 Simulation setup*

vIn order to prove the effectiveness of the improved algorithm, the improved convolutional neural network algorithm proposed in this paper is analyzed and compared with the experimental results of the algorithms in literature [5], literature [6] and literature [7]. In order to guarantee the diversity and authority of data sources, MNIST data set was selected in this paper, which is a handwritten digital database with 60,000 training sample sets and 10,000 test sample sets. We selected 10,000, 20,000 and 30,000 samples as training sets and 2,000 samples as test sets respectively to observe the experimental results.

### *4.2 Results and analysis*

The algorithm in literature [5] is a classical convolutional neural network structure with problems such as too long training time. Compared with the algorithms in literature [5], literature [6] and literature [7], this paper proposes an improved convolutional neural network algorithm and introduces a multi-region logistic regression algorithm to improve the accuracy and training time of image recognition. The comparison between the image recognition rate and the algorithms of literature [5], literature [6] and literature [7] is shown in figure 3. As can be seen from figure 3, with the increase of training times, the image recognition rate of the proposed algorithm has been greatly improved compared with the comparison method, and the advantages are more obvious when the training times are less.

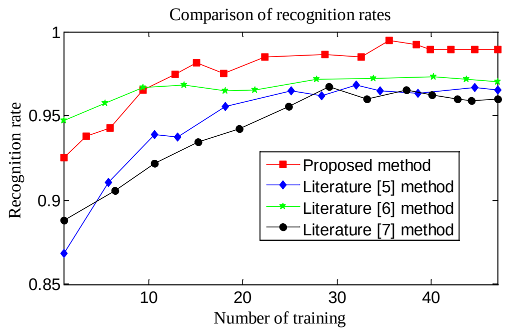


*Figure 3 The comparison of recognition rate*

As the data samples of the training set and the test set are different, the recognition accuracy of the improved convolutional neural network algorithm will vary with the increase of training times. The changes in the accuracy of the proposed algorithm's training set and test set are shown in figure 4.

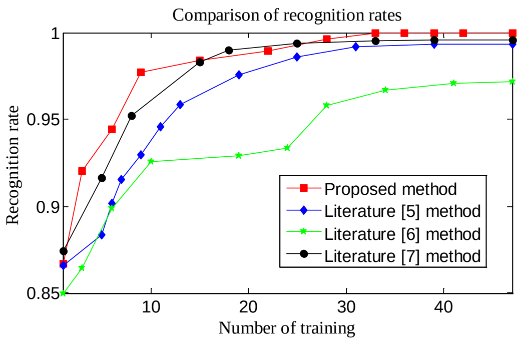


*Figure 4 The comparison of Training and test results*

With the increase of training times, the curve of training time between the proposed algorithm and the algorithms in literature [5], literature [6] and literature [7] is shown in FIG. 5. It can be seen from the figure that the training time of the improved algorithm is significantly shortened.

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

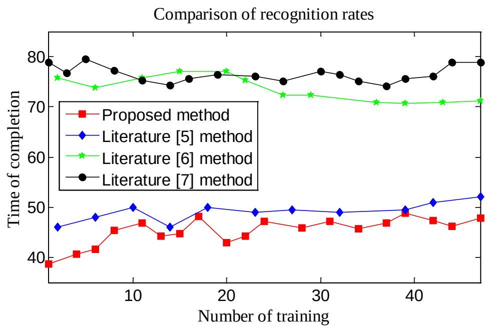


*Figure 5 the comparison of identify time*

We define the method of using multi-region logistic regression calculation as method one and the method of not using multi-region image recognition as method two. In the training stage, due to the different number of training samples, the recognition accuracy is also different. The comparison method and the proposed algorithm are shown in table 1.

*Table 1 the comparison of training results*

| Network type | The number of training samples is 10000 | The number of training samples is 20000 | The number of training samples is 30000 |
|---|---|---|---|
| The recognition rate of literature [5] method | 92.1% | 95.3% | 96.2% |
| The recognition rate of literature [6] method | 92.3% | 95.4% | 96.1% |
| The recognition rate of literature [7] method | 92.1% | 95.1% | 96.7% |
| The recognition rate of the proposed method | **94.5%** | **97.4%** | **98.9%** |

As can be seen from table 1, compared with the comparison method, the recognition accuracy of the proposed algorithm is significantly improved, and with the increase of the number of samples, the accuracy of the proposed algorithm reaches 98.9%. MINIST datasets are used here, and the system saves the error rates of each epoch test. In the training stage, the accuracy of the algorithms in literature [5], literature [6] and literature [7] was 96.2%,96.1%,96.7% when the multi-region image recognition method was not used before by adjusting parameters and changing network structure. After using the multi-region image as the method, the accuracy of the proposed algorithm was about 98.9%, an increase of 2.5%. In the training process, the time of using method 1 is less, but in each test process, the test time is increased due to the increase in the selection of recognition region and the integration process of recognition results. In order to verify this conjecture, method 2 removes the selection process of recognition region based on the original

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

algorithm, and the result pairs are shown in table 2.

*Table 2 the comparison of results*

| Network type | Number of training sample sets | Training time | The test of time | Error rate |
|---|---|---|---|---|
| Comparative approach | 30000 | 78.4s | 104.2s | 3.8% |
| Method 2 | 30000 | 41.8s | **66.2s** | 3.7% |
| Method 1 | 30000 | **39.5s** | 117.6s | **1.1%** |

As can be seen from table 2, compared with the comparison algorithm, method 1 can greatly shorten the training time, and the improved algorithm can greatly improve the accuracy of image recognition, but the total time of image recognition is increased. Method 2 compared with the contrast algorithm, the algorithm without using more area of the image recognition method, the identification accuracy and no obvious change, but greatly reduces the total time of the image recognition, method 2 relative to a, recognition greatly reduces the total time of the same, so method a while can improve the recognition accuracy, but also has its limits, at the expense of a lot of time at the expense of, in the later study, should look for both can effectively combine the way.

## 5. Conclusion

Based on the architecture of typical convolutional neural network and the characteristics of images, a simple convolutional neural network model is constructed in this paper. Because the current data set formats are different, in order to make the system has a certain degree of universality, a method to build data sets meeting the needs of the system is proposed here, which can be used to complete the training process of image classification and view the classification results. In addition, the image set is tested and the results are analyzed by alternating the intra-layer bias sharing and non-sharing of convolution layer. The test results on the MNIST data set show that the proposed method is superior to the comparison algorithm in the recognition rate and other indicators.

## References

[1] Guo L, Li F, Liew W C(2016). Image Aesthetic Evaluation Using Parallel Deep Convolution Neural Network. International Conference on Digital Image Computing: Techniques & Applications.

[2] Lin B Y, Chen C S(2016). Two Parallel Deep Convolutional Neural Networks for pedestrian detection.International Conference on Image & Vision Computing New Zealand.

[3] Chen X, Xiang S, Liu C L, et al(2014). Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. Pattern Recognition.

ISSN 2616-5775 Vol. 2, Issue 3: 34-48, DOI: 10.25236/AJCIS.020305

[4] Gong T, Fan T, Guo J, et al(2016). Gpu-based parallel optimization and embedded system application of immune convolutional neural network. International Workshop on Artificial Immune Systems.
[5] Zhang P, Niu X, Dou Y, et al(2017). Airport Detection on Optical Satellite Images Using Deep Convolutional Neural Networks[J]. IEEE Geoscience & Remote Sensing Letters, vol. 14, no.8, pp.1183-1187.
[6] Yuan X, Pu Y(). Parallel lensless compressive imaging via deep convolutional neural networks[J]. Optics Express, vol. 26, no.2, pp.1962-1977.
[7] Pang S, Yu Z, Orgun M A(2017). A novel end-to-end classifier using domain transferred deep convolutional neural networks for biomedical images[J]. Computer Methods & Programs in Biomedicine, vol.140, pp.283-293.
[8] Li X, Zhang G, Huang H H, et al(2016). Performance Analysis of GPU-Based Convolutional Neural Networks. International Conference on Parallel Processing.
[9] Yang X, Liu C, Wang Z, et al(2017). Co-trained convolutional neural networks for automated detection of prostate cancer in multi-parametric MRI[J]. Medical Image Analysis, vol. 42, pp.212-227.
[10] Lee S, Jha D, Agrawal A, et al(2017). Parallel Deep Convolutional Neural Network Training by Exploiting the Overlapping of Computation and Communication. IEEE International Conference on High Performance Computing.
[11] Hou S, Liu X, Wang Z(2017). DualNet: Learn Complementary Features for Image Recognition. 2017 IEEE International Conference on Computer Vision (ICCV).
[12] Qin M, Xie F, Li W, et al(2018). Dehazing for Multispectral Remote Sensing Images Based on a Convolutional Neural Network with the Residual Architecture. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, vol.99, pp.1-11.
[13] Phan K T, Maul T H, Vu T T(2017). An Empirical Study on Improving the Speed and Generalization of Neural Networks Using a Parallel Circuit Approach[J]. International Journal of Parallel Programming, vol.45, no.4):780-796.
[14] Zhang H, Zhu Q, Jia X(2015). An Effective Method for Gender Classification with Convolutional Neural Networks. International Conference on Algorithms & Architectures for Parallel Processing..
[15] Yakopcic C, Alom M Z, Taha T M(2017). Extremely parallel memristor crossbar architecture for convolutional neural network implementation. International Joint Conference on Neural Networks.
[16] Chu J L, Krzyżak A(2014). Analysis of Feature Maps Selection in Supervised Learning Using Convolutional Neural Networks. Canadian Conference on Artificial Intelligence.
[17] Du L, Du Y, Li Y, et al(2018). A Reconfigurable Streaming Deep Convolutional Neural Network Accelerator for Internet of Things[J]. Circuits & Systems I Regular Papers IEEE Transactions on, vol. 65, no.1, pp. 198-208.
[18] Anthimopoulos M, Christodoulidis S, Ebner L, et al(2016). Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network [J]. IEEE Transactions on Medical Imaging, vol.35, no. 5, pp. 1207-1216.