A Convolutional Neural Network Image Classification Based on Extreme Learning Machine

Shasha Wang, Daohua Liu,Zhipeng Yang, Chen Feng and Ruiling Yao

***Abstract*—To improve the classification accuracy of the massive amount of image data, we propose a novel extreme learning machine classification model for image classification which based on convolutional neural networks. Based on the framework of Alex Net network, the hash function is constructed as the hidden layer between image representation and classification output in convolutional neural network. At the same time, the extreme learning machine is introduced at the end of the network layer, which saves the classification time, improves the classification efficiency and further improves the feature expression ability of the network. Through comparative experiments in the standard databases MNIST and CIFAR-10, the effects of various improved methods under different situations are analyzed. The experimental results show that the proposed convolutional neural network image classification method based on the extreme learning machine improves the average precision by 3%-31% compared with other image classification methods in this paper.**

***Index Terms*—image classification, convolutional neural network, extreme learning machine, hash coding.**

I. Introduction

THE task of image classification is to divide the samples of different categories into different semantic space based on the feature expression ability of the image on the computer, which has been widely used in many fields. Traditional Image classification methods such as content-based Image Retrieval [1] overcome the limitations of manual annotation, but their accuracy depends largely on the accuracy of Image feature extraction. In recent years, neural network has made great progress in vision. The gap between machine extraction of image information features and human understanding of image information is gradually narrowing. For example, VGG with good generalization performance won a good place in the ImageNet image classification competition in 2014 [2]. However, with the increase of global image feature dimension, it is difficult to meet the time requirement by using violent search method. Hash algorithm has been widely used in many visual tasks because of its advantages in computing efficiency and storage. Yang et al. [3] used hash function to match similar data into similar binary code through mapping, which achieved good results in image retrieval performance, but its classification time was long. The subsequent Extreme Learning Machine (ELM) is widely used in image classification due to its random threshold setting, and the connection weight need not be adjusted literately, which improves the speed of training and saves the time of classification. Using extreme learning machine, Liang [4] proposed an algorithm to learn data continuously or one by one. Wei Tao et al. [5] proposed a self-tagging online sequence OS-ELM method, which makes a good improvement on the situation that there are few labeled data in target domain and many unlabeled data in image classification. These improved methods improve the performance of convolutional neural network in image classification task, but there are few methods to introduce hidden layer and hash function which can learn image feature representation on the basis of Alexnet network. The image features and related binary hash coding are learned in a supervised way. The high-dimensional visual data is mapped to the low-dimensional Hamming space by designing hash function, and the classification is reduced by using the improved network structure time-consuming. Experiments show that the CNN image classification method based on ELM has better effects in image classification task, and its performance is better than the existing methods.

Manuscript received June 16, 2020; revised May 7, 2021. This research is supported by the National Natural Science Foundation of China (61402393), the key project of colleges and universities in Henan Province (21A520039) and the research project of teacher education curriculum reform in Henan Province (2021-JSJYZD-030).

Shasha Wang is a teacher of Xinyang Normal University in Henan Province, China. (e-mail: wss1020@126.com).

Daohua Liu is a teacher of Xinyang Normal University in Henan Province, China. (corresponding author, e-mail: ldhzzx @163.com).

Zhipeng Yang is a graduate student of Xinyang Normal University in Henan Province, China. (e-mail: yzp0322@126.com).

Chen Feng is a graduate student of Xinyang Normal University in Henan Province, China. (e-mail: Fch17901@163.com).

Ruiling Yao is a graduate student of Xinyang Normal University in Henan Province, China. (e-mail: yrl_3514@163.com).

II. EXTREME LEARNING MACHINE

Aiming at the shortcomings of traditional single hidden layer neural network, such as long training time and sensitive learning rate, the concept of ELM was proposed. ELM improves the training speed and generalization performance of gradient algorithm. In the process of training, elm not only surpasses the traditional algorithm in learning speed, but also does not need to adjust the parameters of iterative network, so it is widely used in image recognition, speech recognition and other fields. Its structure is as follows: suppose there are N training sample pairs $(\mathbf{x_i}, \mathbf{t_i}), t = 1, 2, 3, ...N$, where, $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$,$t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$ are the corresponding expected output vector, each output is M-dimension, and the dimension of output is its category numb. For a single hidden layer neural network with L hidden layer nodes, it can be expressed as:

$$y_i = \sum_{l=1}^{L} \beta_l g\left(a_l \cdot x_i + b_l\right), i = 1, 2, 3, \cdots N \quad (1)$$

Among them,$a_l = [a_{l1}, a_{l2}, \cdots, a_{lR}]^T$ is the weight vector between the input layer and the L-th hidden layer neurons, b_l is the deviation of the L-th hidden layer neurons ,$\beta_l = [\beta_{l1}, \beta_{l2}, \cdots, \beta_{lM}]^T$ is the weight vector of the network, the objective function is:

$$\sum_{i=1}^{N} \|t_i - y_i\| = 0 \quad (2)$$

Namely:

$$\sum_{l=1}^{L}\beta_l g\left(a_l \cdot x_i + b_l\right) = t_i, i = 1, 2, 3, \cdots N \quad (3)$$

The form of matrix is as follows:

$$H\beta = T \quad (4)$$

Among them, H is the output matrix , β is the output weight, T is the expected output.

$$\beta = [\beta_1, \beta_2, \cdots, \beta_L]^T, T = [t_1, t_2, \cdots, t_N]^T \quad (5)$$

$$H = \begin{bmatrix} h\left(x_1\right) \\ \vdots \\ h\left(x_N\right) \end{bmatrix} = \begin{bmatrix} g\left(a_1, b_1, x_1\right) & \cdots & g\left(a_L, b_L, x_1\right) \\ \vdots & \ddots & \vdots \\ g\left(a_1, b_1, x_N\right) & \cdots & g\left(a_L, b_L, x_N\right) \end{bmatrix} \quad (6)$$

Since the connection weights between input layers are set randomly in the learning process of ELM, and the weights between output layers can be determined by solving equations. Therefore, the calculation is relatively simple, and the connection weightβ can be calculated according to formula (4).

III. BINARY HASH CODING LEARNING OF HIDDEN LAYER

Supervised hash learning is a process in which the category labels of images are used to estimate the output of binary coding model, and then the model parameters are adjusted continuously according to the errors between the actual output and the expected output in the iterative process. Based on the revelation in literature [6], this paper constructs a hidden layer between the network output layers, and its activation function is:

$$a_n^H = \sigma\left(a_n^7 w^H + b^H\right) \quad (7)$$

In the above equation, $w^H \in R^{d\times k}$ represents the network weight of this layer , a_n^H is the K-dimensional vector, b^H is the bias term, $\sigma()$ is the Sigmoid logic function, where $\sigma(z)$ is defined as $\sigma(z) = \frac{1}{(1+\exp(-z))}$, and the binary coded function is given by the following formula:

$$b_n = \left(\operatorname{sgn}\left(a_n^H - 0.5\right) + 1\right)/2 \quad (8)$$

When the network framework is constructed, the hash learning method is used to index the data, which can not only preserve the tag semantics between images, but also improve the ability of image classification. At the output end of the convolutional neural network, the symbol function $\operatorname{sgn}()$ is used to quantify the quasi-hash code into the binary hash code. Given data set $\chi = \{x_1, x_2, x_3, \cdots x_n\}$, d is dimensions of data points, using kernel function $\kappa : R^d \cdot R^d \to R$ to construct hash function that satisfy $R^d \to \{0, 1\}$. Inspired by Kernelized Locality-Sensitive Hashing KLSH [7] algorithm, a hash function using kernel function κ is defined, and the formula is as follows:

$$f(x) = \sum_{j=1}^{m}\kappa\left(x_{(i)}, x\right)a_j - b \quad (9)$$

$$h\left(f(x)\right) = \operatorname{sgn}\left(f(x)\right) = \begin{cases} 1, & f(x) > 0 \\ 0, & f(x) \le 0 \end{cases} \quad (10)$$

Among this, $x_{(i)}$ is a randomly selected sample from χ, $a_j \in R$,$a_j \in R$. The design of hash function should not only obtain enough information, but also satisfy the balance as much as possible, it should satisfy $\sum_{i=1}^{n} h\left(x_i\right) = 0$, bias term $b = \sum_{i=1}^{n}\sum_{j=1}^{m}\kappa\left(x_{(j)}, x_i\right)a_j/n$, bring bias term b into formula (9) and get $f(x) = \sum_{j=1}^{m}\left(\kappa\left(x_{(j)}, x\right) - \frac{1}{n}\sum_{i=1}^{n}\kappa\left(x_{(j)}, x_i\right)\right)a_j = a^T\overline{k}(x)$, among this $a = [a_1, \cdots, a_m]^T$, $\overline{k} : R^d \to R^m$ is mapping vector:

$$\overline{k}(x) = \left[\kappa\left(x_{(1)}, x\right) - \mu_1, \cdots, \kappa\left(x_{(m)}, x\right) - \mu_m\right]^T \quad (11)$$

Among this $\mu_j = \sum_{i=1}^{n}\kappa\left(x_{(j)}, x_i\right)/n$ can be calculated in advance.

IV. A CONVOLUTIONAL NEURAL NETWORK IMAGE CLASSIFICATION BASED ON ELM

A. Conventional Convolutional Neural Network

CNN operates the input image layer by layer, from extracting low-level features that can represent the basic attributes of the image to inferring more advanced image features, which shows that the convolutional neural network has strong recognition and representation capabilities. The convolutional layer uses a convolution core of a certain size to act on the local image region to extract image features. The pooling layer is to down sample the convoluted output image by imitating the human visual system. The fully connected layer maps the learned hidden layer features to the label space of the sample. The training process of the network is as follows:

1) Forward propagation stage: The operation of each layer input feature is as follows:

$$y^{(l)} = f\left(\sum_{i\in m} W_i^l \otimes x_i^{(l-1)} + b^l\right) \quad (12)$$

Among this, l is the number of layers, $y^{(l)}$ is the result of the output layerx_i is the input, $\otimes$ is the convolution calculation, and b^l is the offset term. f represents the activation function, m is the input feature set and W is the weight of the layer.

2) Back Propagation Stage: Back propagation adjusts the parameters according to the error until the expected output is achieved:

$$\arg\min_W E(W) = \arg\min_W \sum_{i=1}^{M} L\left(z_i\right) + \lambda\|W\|^2 \quad (13)$$

Among this, $L()$ denotes the loss function, W denotes the weight of the network, λ is the regularization term, z_i is the input of the network. The cross-entropy loss function is calculated based on SoftMax, which converts the final output of the network into a probability function form through exponential transformation. The loss function is defined as follows:

$$L\left(z_i\right) = -\log\sigma_i(z) \quad (14)$$

$$\sigma_i(z) = \frac{\exp\left(z_i\right)}{\sum_{j=1}^{m}\exp\left(z_j\right)}, i = 1, 2, 3, \cdots, m \quad (15)$$

$$z_i = z_i - \max\left(z_1, z_2, z_3, \cdots, z_m\right) \quad (16)$$

Among this, z_i is the predicted output, and $\sigma_i(z)$ predicted input value belongs to the probability value of a certain category.

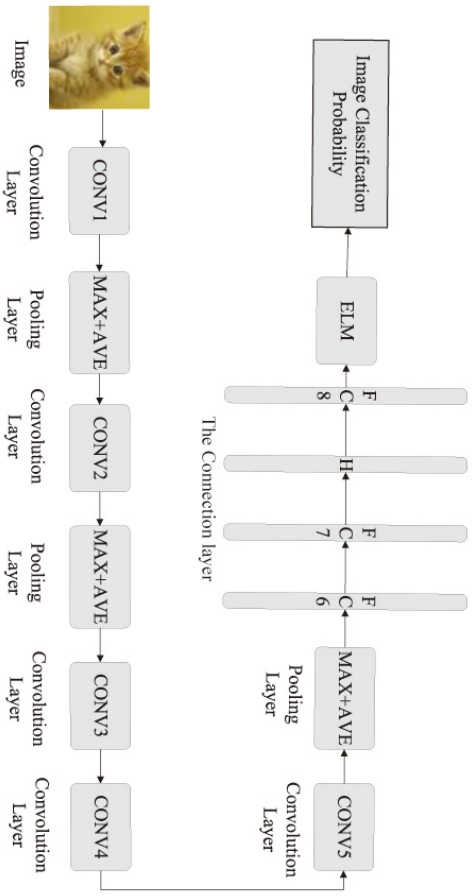

Fig. 1. AlexNet Network Model Framework Optimization Diagram. This diagram describes the improved network framework in detail.

B. A Convolutional Neural Network Image Classification Method Based on Extreme Learning Machine

The proposed improved method is AFCME (AlexNet+FC+Max-Ave-Pooling+ELM). The input data are images and corresponding labels, and the output result is the probability of the image belonging to a certain category. The improved framework is shown in Figure 1. The network framework is improved as follows:

1) The method of max-ave pooling is used to replace the maximum pooling and expand the local receptive field;

2) The hash function is constructed on the full connection layer as the hidden layer between image representation and classification output;

3) ELM is introduced into the full connection layer to reduce the classification time and improve the classification ability of the network structure.

The improved network model can be divided into two stages: training and testing:

Input: training sample data set and expected output $(x_i, t_i), i = 1, 2, 3, \cdots N$.

Output: Trained image classification network model.

step1. Initialization of AlexNet network parameters, target accuracy ε and the maximum number of iterations allowed e.

step2. When the number of iterations $<$ e&err $>$ ε , the actual output is calculated according to $y_i = \sum_{l=1}^{L} \beta_l g\left(a_l \cdot x_i + b_l\right), i = 1, 2, 3, \cdots N$.

step3. Random gradient descent method is used to adjust the corresponding parameters:

step4. Find out the average value of current batch data x_i ,$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i$;

step5. Calculate the variance of current batch normalized processing data size ${\sigma_\beta}^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \mu_\beta\right)^2$;

step6. x_i is normalized and $\bar{x}_i$ is obtained;

step7. The scaling and translation variables γ and β are introduced to calculate the normalized value $y_i = \gamma \bar{x}_i + \beta$.

step8. Random generation of extreme learning machine parameters W and b .

step9. The reciprocal second layer of convolutional neural network is calculated according to $y_i = \sum_{l=1}^{L} \beta_l g\left(a_l \cdot x_i + b_l\right), i = 1, 2, 3, \cdots N$ and used as the input of the extreme learning machine layer.

step10. Calculate H and β according to $H\beta = T, \beta = \left[\beta_1, \beta_2, \cdots, \beta_L\right]^T$.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The experiment uses Caffe in the Windows system to implement the improved network framework shown in Figure 1. The experiment is configured with Intel Core i7-6700 and the CPU main frequency is 3.41GHz. In this paper, the open source Caffe framework is used to implement the proposed network framework. At the same time, the experimental data is batch standardized.

Two common data sets MNIST and CIFAR-10 are used to verify the effectiveness of the proposed improved method. In order to verify the effectiveness of the improved method, AlexNet+FC (AFC) proposed in this paper adds a hidden layer on the basis of AlexNet model. The Normalization (AFCB) method of AlexNet+FC+Batch Normalization (AFCB) is based on the Normalization of data in hidden layers. The maximum-mean Pooling method was used to replace the maximum-pooling method in the experiment, and AlexNet+FC+max-ave-pooling (AFCM) method was used for verification. During the experimental process, three methods with good performance are selected for comparison experiments with AFCME, such as Stochastic Pooling [8] (SP, NIN+Dropout[9] (ND) and AlexNet+Fine-tuning[3] (AF). In this paper, the evaluation index of image classification performance is evaluated by the relationship between the mean value of error rate, precision ratio and average accuracy and the number of hidden layer neurons.Formula 17 is the precision rate; Formula 18 is the MAP. Among them, N_r is the number of similar images, N_t denotes the number of all images which included in the classification results , N_K is the number of samples accurately classified, $p\left(k\right)$ denotes the accuracy of k position , $rel\left(k\right)$ denotes whether k position is related to the classification category, and N_{q} is the total number of samples.

$$precision = \frac{N_r}{N_t} \tag{17}$$

$$\mathrm{MAP} = \frac{1}{N_{\mathrm{q}} \cdot N_K} \sum_{q=1}^{N_q} \sum_{k=1}^{N_k} p\left(k\right) \times rel\left(k\right) \tag{18}$$

MNIST data set is generally divided into 10 categories, so the channel of network output image is set as 10, and the data is trained by stochastic gradient descent method. In the experiment, several methods with better performance were selected to compare the accuracy with MAP. The results are shown in Fig.2 and Fig.3.

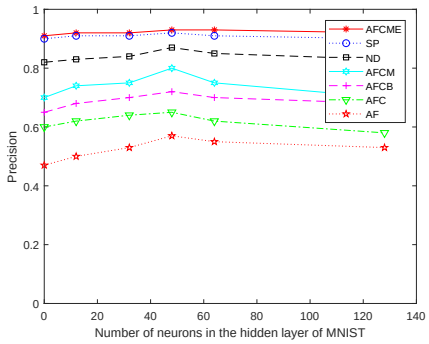

Fig. 2. Accuracy comparison chart. This figure describes the trend of the change of precision rate with the number of hidden layer neurons in MNIST data set.

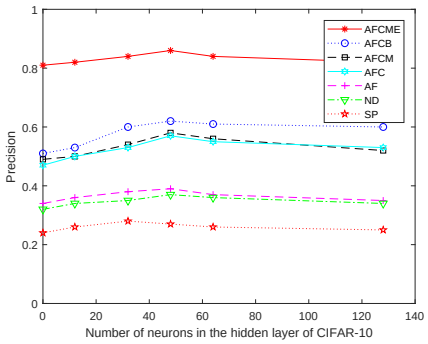

Fig. 4. Accuracy comparison chart. This figure describes the trend of the change of precision rate with the number of hidden layer neurons in CIFAR-10 data set.

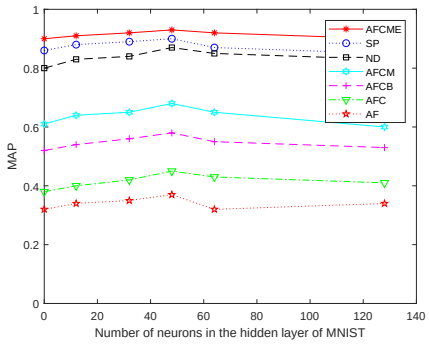

Fig. 3. Average Precision Mean Contrast Diagram. This figure describes the comparison diagram of the relationship between the number of hidden layer neurons and the mean precision mean on the MNIST data set.

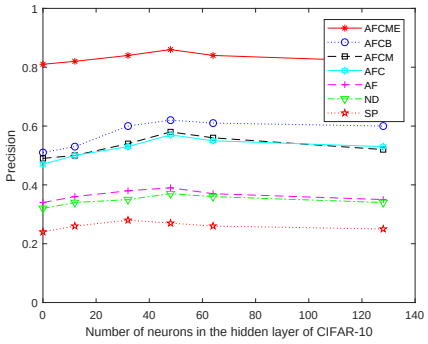

Fig. 5. Average Accuracy Mean Contrast Diagram. This figure describes the comparison diagram of the relationship between the number of hidden layer neurons and the mean precision mean on the CIFAR-10 data set.

Figure 2 shows the precision comparison chart of the proposed method and the comparison method on MNIST dataset under different hidden layer nodes. It can be seen from the figure that the precision of AFCME is not very different from the current better performance methods, such as NIN+Dropout and AlexNet+Fine-tuning .But when the number of hidden layer neurons is 48, the precision of the proposed method reaches 92% .The reason is that the MNIST dataset contains a relatively single image type, and the image feature extraction is relatively simple. After several convolution layers, the main information is extracted.

Figure 3 shows the comparison of average precision mean values of AFCME and Stochastic Pooling, NIN+Dropout and AlexNet+Fine-tuning on MNIST dataset. When the number of hidden layer neurons is 48-64, the MAP values of AFCME are 3% higher than the Stochastic Pooling values, indicating that the proposed method, compared with the contrast method, has the characteristics of high average accuracy and stable performance. The reason is that MNIST feature extraction is relatively easy, blindly increasing the number of neurons will increase the test error rate, which will make the generalization ability of the model too poor. On the contrary, in the later stage, the classification accuracy of the network basically remains unchanged.

CIFAR-10 data set contains 10 types of images, so in order to match the probability of image output from the network, set the output channel of the network as 10. The same as MNIST data set, the experiment adopts the same comparison method, and compares the precision and average precision with the proposed method. The results are shown in Fig.4 and Fig.5.

Figure 4 shows the precision comparison chart of the proposed method and the comparison method on CIFAR-10 dataset under different hidden layer nodes. The experimental results in Figure 4 show that AFCM has better generalization and classification performance compared with current image classification methods. And by connecting the ELM at the end of the network layer, the precision can reach 78%when the number of neurons is 64, which shows that the improved method proposed in this paper has good optimization. It can be seen from the curve that the results of AFCME on the CIFAR-10 data set are characterized by high precision and stable performance compared with the existing methods, which indicating that the improved method is more effective for image feature extraction.

Figure 5 shows the comparison of average precision mean values of AFCME and Stochastic Pooling, NIN+Dropout and AlexNet+Fine-tuning on CIFAR-10 dataset. Experimental results show that the proposed method in combination with the largest - average on the basis of their respective advantages not only in the process of feature extraction to keep the translation invariance of the deformation and small deformation, and combined with the advantages of extreme learning machine using the generalized inverse calculation, reduce the classification of consumption, accelerate the training speed of the network, and therefore is good method to do contrast experiment has better performance.

In the experiment of the error rate of image classification

TABLE I
ERROR CLASSIFICATION RATE OF CNN MODEL ON MNIST DATASET

Method	Error rate (%)
Stochastic Pooling [8]	0.49
NIN+Dropout [9]	0.47
AlexNet+Fine-tuning [3]	0.64
AlexNet+FC	0.66
AlexNet+FC+Batch Normalization	0.63
AlexNet+FC+Max-Ave-Pooling	0.52
AlexNet+FC+Max-Ave-Pooling+ELM	0.50

TABLE II
ERROR CLASSIFICATION RATE OF CNN MODEL ON CIFAR-10 DATASET

Method	Error rate (%)
Stochastic Pooling [8]	15.51
NIN+Dropout [9]	10.67
AlexNet+Fine-tuning [10]	12.64
AlexNet+FC	11.53
AlexNet+FC+Batch Normalization	11.13
AlexNet+FC+Max-Ave-Pooling	10.14
AlexNet+FC+Max-Ave-Pooling+ELM	9.85

performance, the comparison results of the error rates of the following different classification methods are obtained. The experimental results of the two datasets are shown in Table 1 and 2.

Analysis of the data in Table 1 shows that the error classification rate of the proposed method can reach 0.5%. The error rate for batch normalization used is 0.63% due to the fact that Batch Normalization is generally used in frameworks with more complex network structures. The error rate of the improved network framework can reach a small value when the number of training iterations is very small. In this way, even if ELM is accessed on the basis of hidden layer, it will not give full play to its advantage.

Table 2 shows the error classification rate of CNN model on CIFAR-10 dataset. Adding hidden layer alone can not achieve the maximum performance improvement, and can only reach the general level in all methods. However, the error classification rate of the model is significantly reduced to 9.85% by using maximum-mean pooling instead of the original pooling method. The reason is that the new pooling method is more conducive to the classification and recognition of images.

In order to verify the influence of the improved network framework on the calculation time, one of the CIFAR-10 data sets was selected in the experiment to compare the training time spent by AFCME with Inception-v4 and AlexNet+Fine-tuning methods. The experimental results are shown in Table 3. The total duration of one-time forward propagation and backward propagation in the network training process was selected as the basis for comparison. Compared with the traditional CNN, the training time is shortened to 0.09s because the Extreme Learning Machine adopts the property of finding generalized inverse for feature classification, which indicates the effectiveness of the proposed improved method.

TABLE III
COMPARISON OF TRAINING DURATION BETWEEN FORWARD AND REVERSE PROPAGATION

Method	Training time(s)
AlexNet+Fine-tunin[3]	0.126
AlexNet+FC	0.164
Inception-v4[3]	0.124
AlexNet+FC+Max-Ave-Pooling	0.146
AlexNet+FC+Max-Ave-Pooling+ELM	0.09

VI. CONCLUSIONS

In this paper, an improved CNN image classification method is proposed. This method uses the hidden layer connected by AlexNet to improve the accurate expression of learning features, and at the same time uses the extreme learning machine introduced after full connection layer to shorten the calculation time and improve the accuracy of image classification. The experimental results show that the proposed method can extract more distinguishing and expressive deep features. However, there are still some improvements in this method. When the degree of discrimination of extracted feature is not obvious, the accuracy of classification results cannot achieve the desired results. In the future, the framework of the network can be further studied to further reduce the error rate of image classification, improve the expression ability of learning features and shorten the calculation time.

REFERENCES

[1] Y. Gong and S. Lazebnik, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 35, pp. 2916-2929, 2013.

[2] J. Redmon and S. Divvala, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788.

[3] H. Yang and K. Lin, "Supervised Learning of Semantics-Preserving Hash via Deep Convolutional Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 437-451, Feb.2018.

[4] N. Y. Liang and G. B. Huang "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks," *IEEE Transactions on Neural Network*, vol. 6, no. 17, pp. 1411-1423, 2006.

[5] T. Wei and X. S. Ji "Image Recognition Method Based on Self-labeling Online Sequential Extreme Learning Machine," *Computer Engineering*, vol. 6, no. 42, pp. 208-212, 2016.

[6] C. Bai and L. S. Huang "Optimization of deep convolutional neural network for large scale image classification," *Journal of Software*, vol. 4, no. 29, pp. 1029-1038, 2018.

[7] S. F. Chang and Y. G. Jiang, "Supervised hashing with kernels," in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence*, pp. 2074-2081.

[8] M. D. Zeiler and R. Fergus "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks," *Eprint Arxiv*, vol. 37, no. 1, pp. 34-39, 2012.

[9] G. E. Hinton and N. Srivastava "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 4, no. 3, pp. 212-223, Jul. 2012.

[10] J. Deng and W. Dong, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 207-216.