Computer Science and Information Systems 20(4):1771–1795 https://doi.org/10.2298/CSIS221210030D

A Novel Multilevel Stacked SqueezeNet Model for Handwritten Chinese Character Recognition

Yuankun Du[1], Fengping Liu[2], and Zhilong Liu[3]

[1] School of Big Data and Artificial Intelligence, Zhengzhou University of Science and Technology
450002 Zhengzhou, China
duyuankk@163.com

[2] School of Information Engineering, Zhengzhou University of Science and Technology
450002 Zhengzhou, China
liufengppp@163.com

[3] Library of Henan Institute of Animal Husbandry and Economics
450002 Zhengzhou, China
liuzhilonglo@163.com

Abstract. To solve the problems of large number of similar Chinese characters, difficult feature extraction and inaccurate recognition, we propose a novel multilevel stacked SqueezeNet model for handwritten Chinese character recognition. First, we design a deep convolutional neural network model for feature grouping extraction and fusion. The multilevel stacked feature group extraction module is used to extract the deep abstract feature information of the image and carry out the fusion between the different feature information modules. Secondly, we use the designed down-sampling and channel amplification modules to reduce the feature dimension while preserving the important information of the image. The feature information is refined and condensed to solve the overlapping and redundant problem of feature information. Thirdly, inter-layer feature fusion algorithm and Softmax classification function constrained by L2 norm are used. We further compress the parameter clipping to avoid the loss of too much accuracy due to the clipping of important parameters. The dynamic network surgery algorithm is used to ensure that the important parameters of the error deletion are reassembled. Experimental results on public data show that the designed recognition model in this paper can effectively improve the recognition rate of handwritten Chinese characters.

Keywords: Handwritten Chinese character recognition, multilevel stacked SqueezeNet model, inter-layer feature fusion, L2 norm.

1. Introduction

Handwritten Chinese character recognition (HCCR) is one of the most challenging problems in pattern recognition and machine learning [1,2]. Optical Character Recognition (OCR) involves many disciplines such as digital signal processing, pattern recognition and natural language processing, and it has been widely applied in computer and other related fields [3]. Handwritten Chinese character recognition being realized can be used for machine marking, mail automatic sorting, bill recognition, etc,. However, there are a large number of similar Chinese characters due to various categories of Chinese characters, complex font structure [4]. Handwritten Chinese characters are different from person to person, resulting in handwritten Chinese character recognition difficulties, so handwritten Chinese character recognition has been a research difficulty and hotspot.

According to the collection method of handwritten Chinese characters data, it can be divided into off-line handwritten Chinese characters recognition and online handwritten Chinese characters recognition. The off-line handwritten Chinese character pictures are captured by cameras or scanners and other instruments [5]. Online handwritten Chinese character recognition collects handwritten Chinese characters by various hardware devices in real time. In this process, not only the characteristics of Chinese characters, but also the stroke track information of Chinese characters are collected [6]. Off-line handwritten Chinese characters inevitably add noise interference in the process of picture acquisition. Therefore, in general, off-line handwritten Chinese character recognition is more difficult than online handwritten Chinese character recognition. Traditional off-line handwritten Chinese character recognition mainly includes three steps: data preprocessing, feature extraction and recognition classification. Among them, data processing mainly involves smoothing and de-noising, whitening, shaping and transforming. Feature extraction mainly includes statistical features and structural features. The statistical features have better effects than structural features, which mainly include Gabor feature [7] and Gradient feature [8], etc. Support vector machine classifier and linear discriminant classifier are mainly used to identify the differences.

In recent years, the traditional ”pre-processing+feature extraction+classifier” handwritten Chinese character recognition does not seem to have made great progress, and there are few breakthrough research reports. However, the rise of deep learning has brought new vitality and extremely effective solutions to handwritten Chinese character recognition problems, especially the introduction of Convolutional Neural Network (CNN), which makes breakthroughs in the field of image recognition. Deep convolutional neural network models such as VGGNet, improved Inception, ResNet and other models have achieved excellent results on the ImageNet data set [9]. These advanced technologies provide the basis and reference for off-line handwritten Chinese character recognition.

So far, many researchers have done a lot of researches on off-line handwritten Chinese character recognition. Some researchers conduct research based on traditional machine learning methods. For example, reference [10] adopts an improved affine propagation clustering algorithm. In reference [11], a multi-feature handwritten Chinese character recognition technology based on support vector machine SVM was proposed. On the basis of grid feature extraction, centroid features, stroke features and feature points of Chinese characters were extracted, and SVM classifier was adopted to realize handwritten Chinese character recognition. This kind of method requires data preprocessing and complex feature extraction, so it is difficult to extract accurate features comprehensively. In reference [12], a partial cascade feature classification scheme based on LS-SVM was adopted. The sampling results of low-threshold Hough space were used as coarse classification features, and the local two-branch distribution histogram was used as fine classification features for coarse classification. Sample classification was realized after coarse classification. Reference [13] used Modified Quadratic Discriminant Function (MQDF) and the Convolutional Neural Networks (CNN) to obtain a higher accuracy than the single CNN and MQDF. Reference [14] adopted the cascaded MQDF and Deep Belief Networks (DBN) to achieve higher accuracy. Based on deep learning algorithms, some researchers improve the recognition performance by improving the network structure or proposing improved training methods. For example, Residual Networks (ResNet) were used in reference [15] to optimize network performance by improving the unit structure of residual learning module. In reference [16], iterative refinement was adopted in convolutional neural networks. Reference [17] applied the VGGNET model in convolutional neural network to Chinese character recognition. In reference [18], the center loss function proposed in face recognition was applied to the CNN network of handwritten Chinese characters to reduce the intra-class distance, increase the inter-class distance and improve the recognition performance. Wang et al. [19] used a deep CNN network, combined the printed and handwritten data sets to train the recognition network, and builded a service to expand the training data set and improved the adaptability to different writing styles.

Although the recognition accuracy of handwritten Chinese characters based on CNN model has been greatly improved, it requires large computing resources, power consumption and storage space, it has many parameters, and it is difficult to conduct distributed training. It is the greatest challenge for deploy the corresponding model in embedded platforms such as ARM board and FPGA with limited hardware resources [20]. In order to realize handwritten Chinese character recognition with limited resources, the size of the model is reduced as much as possible while the model prediction performance is guaranteed in this paper.

There are five commonly used methods to compress CNN model volume: network pruning, parameter sharing, quantization, network distillation and compact network design, all of them can obtain obvious compression effect. The compact network improves the convolution with more network parameters and computations. For example, SqueezeNet, ShuffleNet, MobileNet, and Xception all have reduced the convolution layer [21].

In the proposed multilevel stacked SqueezeNet, FireModule is introduced into AlexNet convolutional model, and the model is compressed 50 times with good accuracy, and is successfully applied to embedded platform. In this paper, the compression of handwritten Chinese character model is studied. After modifying the SqueezeNet model, the Dynamic Network Surgery model is added to compress the parameters of the model, including cutting and repairing, and the accuracy of the model is ensured at the same time.

This part is the organization structure. Section 2 introduce the related works of CNN. Section 3 shows the detailed multilevel stacked SqueezeNet model. The experiments and analysis are conducted in section 4. Section 5 concludes this paper.

2. Related Works

Convolutional neural network (CNN) is a structured and supervised multi-layer feed-forward neural network, which is mainly applied to two-dimensional data processing. It can automatically extract image features through learning, and finally achieve the purpose of image classification. It consists of alternating convolution layers, sampling layers and fully connection layers, each of which contains multiple convolution kernels. Each neuron in the convolutional layer is connected with the local region of the upper layer, and the feature information of two-dimensional data is extracted through the convolutional operation, and the interference of noise on the feature is reduced. The sampling layer samples two-dimensional data, reduces the resolution, saves the feature information of the image as much as possible, reduces the dimension of the data and the number of parameters, then it improves the network operation speed. LetNet-5 is a classical convolutional neural network structure [22], and its structure diagram is shown in Figure 1.

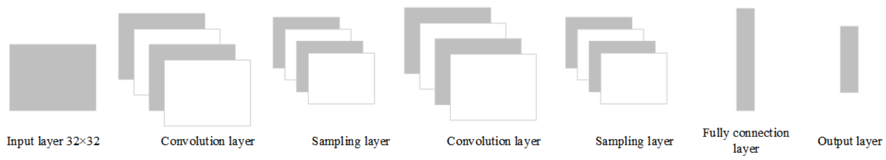

Fig. 1. Structure of LetNet-5

2.1. Convolutional Layer

For the convolutional layer, the convolutional kernel performs sliding convolution operation with the feature map of the previous layer, and the bias is added to get a net output, as shown in Formula (1). Finally, the result of convolution is obtained through the nonlinear action of activation function, namely, the output feature map, i.e.,

$$u_i^l = \sum_j^M x_j^{l-1} \times k_{ij}^l + b_i^l, i = 1, 2, \cdots, N \tag{1}$$

$$x_i^l = f(u_i^l) \tag{2}$$

Where, N represents the number of convolution kernels in the $l-th$ layer. M represents the number of feature maps at layer $l-1$. x_j^{l-1} is the $j-th$ feature graph in layer $l-1$. k_{ij}^l is the $j-th$ channel in the $i-th$ convolution kernel of the l layer. b_i^l is the bias of the $i-th$ convolution kernel. u_i^l is the net output of the $i-th$ convolution kernel in layer 1. The operation $\times$ stands for convolution operation. x_i^l is the $i-th$ feature graph of the $l-th$ layer. $f(\cdot)$ is the activation function, usually Sigmoid function. The Sigmoid function is shown in Formula (3):

$$S(x) = \frac{1}{1+e^{-x}} \tag{3}$$

2.2. Sampling Layer

In the sampling layer, the output feature map of the previous layer is down-sampled, and the input feature map is divided into multiple non-overlapping image blocks by sampling window, and then the maximum pooling or average pooling method is adopted for each image block. Assume that the size of the sampling window is $n \times n$, the size of the input feature graph is $iS \times iS$, and the size of the output feature graph is shown in Formula (4). Maximum pooling and average pooling are shown in Equation (5) and Equation (6).

$$oS = \frac{iS}{n} \tag{4}$$

$$x_{kj}^{l} = max(x_{k1}^{l-1}, \cdots, x_{kn^2}^{l-1}), x_{ki}^{l-1} \in V_{kj}, k = 1, 2, \cdots, N \tag{5}$$

$$x_{kj}^{l} = \frac{1}{n \times n} \sum x_{ki}^{l-1} \tag{6}$$

Where l represents the current sampling layer. N represents the number of input feature maps, which is the same as the number of output feature maps. $V_{kj}, j = 1, 2, \cdots, oS^2$ represents the $j-th$ image block of the $k-th$ input feature graph. Each image block contains n^2 elements. x_{ki}^{l} is the $i-th$ element in V_{kj} image block. x_{kj}^{l} is the $j-th$ element on the $k-th$ output feature map of the current layer.

2.3. Fully Connection Output Layer

After the convolution layer and the down-sampling layer, the advanced features of the original image have been extracted. The purpose of the fully connection layer is to use these features to classify the original image. The fully connection layer does the weighted sum of these features, adds the bias quantity, and finally obtains the final output by activating the function, as shown in Equation (7). The output layer is also essentially a fully connection layer, except that the activation function is classified by the classification function [23].

$$y^{l} = f(w^{l} x^{l-1} + b^{l}) \tag{7}$$

Where, x^{l-1} is the output feature diagram of the previous layer, and the elements are high-level features extracted through convolution and down-sampling. w^{l} is the weight coefficient of the fully connection layer. b^{l} is the offset of the fully connection layer l.

The convolutional neural network is generally used in the case of multiple classifications, and the classification function usually adopts Softmax function, which normalizes the inactive output Z^{L} of the last layer L to the range of $(0, 1)$. Meanwhile, the sum of output values is 1 for classification. The calculation formula of Z^{L} is shown in Equation (8), and the function of Softmax is shown in Equation (9).

$$z^{L} = w^{L} x^{L-1} + b^{L} \tag{8}$$

$$a_{i}^{L} = \frac{e^{z_{i}^{L}}}{\sum_{j=1}^{n^L} e^{z_{i}^{L}}} \tag{9}$$

Where L represents the last output layer ($L - th$ layer). n^L denotes that there are n output neurons in layer L. a_i^L is the output of the $i - th$ category in n categories. z_i^L represents the $i-th$ inactive output. z_j^L represents the $j-th$ inactive output. e is a natural constant.

2.4. SqueezeNet Model

SqueezeNet is based on AlexNet model. It is with fewer parameters while achieving an accuracy close to AlexNet network [24]. The core of SqueezeNet lies in FireModule, where small convolution kernels replace partially large ones. When 5×5 and 3×3 convolutions are used for convoluted one $5 \times 5 \times 1$ image, the former will produce 25 parameters and 25 calculations, while the latter will produce 18 parameters and 90 computations. However, the memory reading speed of the computer is much slower than that of multiplication calculation, and the convolution speed of small convolution kernel with fewer parameters is faster. Therefore, in this paper, 1×1 is used to replace part 3×3, which will accelerate the convolution speed, and the remaining 3×3 convolution kernel guarantees the convergence speed. the convolution speed. The other 3×3 convolution kernels ensure the convergence speed.

As shown in Figure 2, SqueezeNet is divided into Squeeze layer and Expand layer. The Squeeze layer has S convolution layers with 1×1 kernel. The Expand layer is a convolution layer with e_1 1×1 and e_2 3×3 convolution kernels, and the activation layer is ReLU. The input feature map size of FireMoudle is $H \times W \times M$, and the output feature map size is $H \times W \times (e_1 + e_2)$. Only the dimension is changed, but its resolution is not changed. Firstly, the feature maps of $H \times W \times M$ are squeezed through the Squeeze layer, and S feature maps are obtained to achieve compression effect ($S < M$). In the Expand layer, $H \times W \times S$ is convolved with e_1 1×1 convolution kernels and e_2 3×3 convolution kernels respectively, and the convolution results of the two parts are fused to obtain the output result with $H \times W \times (e_1 + e_2)$ size. The value of $(e_1 + e_2)$ must be greater than M. So FireMoudle increases the dimension of the input. Where, S, e_1, and e_2 are adjustable parameters, which represent the number of convolution kernels and also reflect the dimension of the output feature graph. In this paper, $e_1 = e_2 = 4S$ is taken.

In addition, down-sampling operation is used in the module to ensure that the convolutional layer has a larger activation function, and the model accuracy is guaranteed under the condition of limited network parameters.

3. Multilevel Stacked SqueezeNet Model

Traditional convolutional neural networks have some problems, such as inadequate feature extraction and poor network learning ability. At the same time with the deepening of the network, it also has the information loss issue. In order to solve these problems, this paper uses the advantages of ResNet residual network to transfer information directly to the output results, thus protecting the integrity of information and alleviating the problem of information loss. The feature information is divided into groups, and then feature extraction is carried out respectively. Finally, the information of each group is integrated together to increase the diversity of online learning.

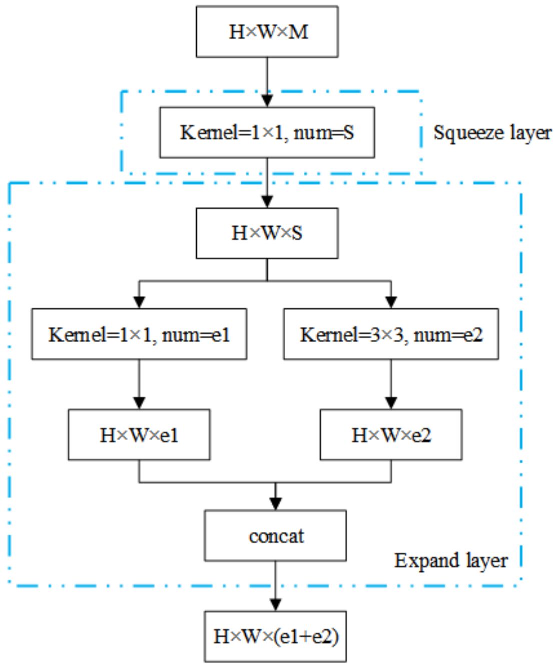

Fig. 2. FrieMoudle module

3.1. Feature group extraction module

The feature group extraction module designed in this paper is shown in Figure 3 and Figure 4. The number of input feature maps of the network module in Figure 3 is 129. Before the feature grouping, channel rearrangement of the feature information is performed to disrupt the order of input feature information.

As shown in Figure 5, taking three groups as an example, in order to increase the diversity of network learning, the feature information is grouped with different colors to represent different information. In the absence of channel rearrangement, the same feature information may be contained in the same group segment, but the information in different groups segment will be different. If features are directly grouped, the information in the group segmentation will be incomplete and the representation ability of the information will be reduced. It can be seen from the figure that rearrangement of channels enables different segments to exchange information, enriching the feature information of different segments. Through channel rearrangement, each group has the characteristic information of other groups. In this way, although there is no contact between groups after grouping, the information of each group is comprehensive and will not be lost. Then, 129 feature graphs are divided into three groups with 43 features in each group. In order to enhance the diversity of network extraction information, convolution kernels of different sizes are used for feature extraction in each group. After information extraction for each group, information exchange and fusion between groups are carried out. At this time, the number of feature maps for each group increased from 43 to 86. Then 86 feature maps of each

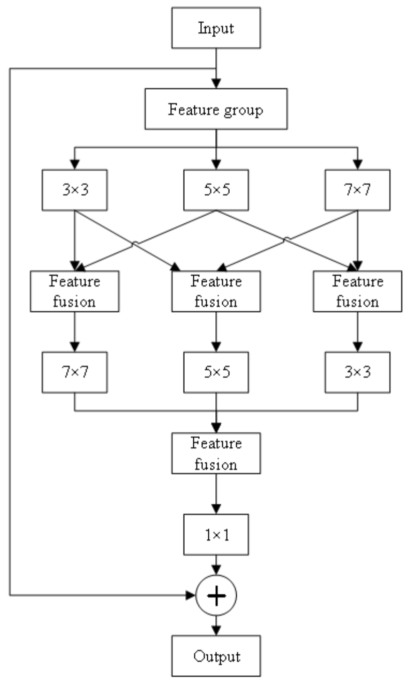

Fig. 3. Network module 1

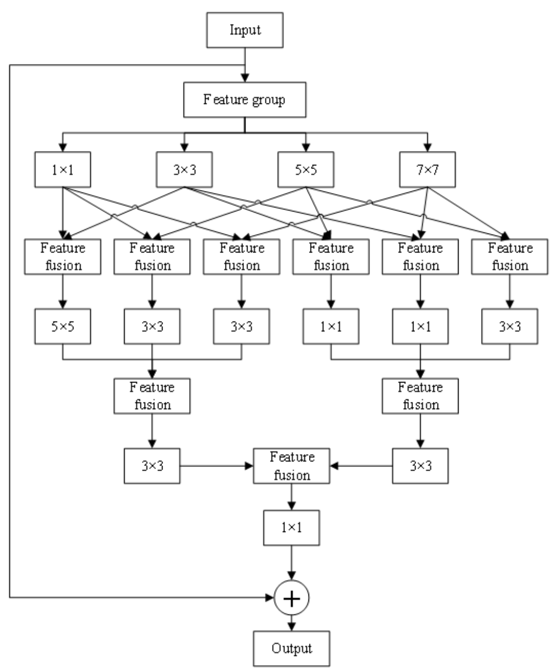

Fig. 4. Network module 2

group are extracted to further improve the learning ability of the network. Finally, the information of each group is merged and integrated, and the number of integrated feature channels is 258. In order to make them consistent with the input channel and facilitate the residual calculation with the original input information, 1×1 convolution is needed to reduce the dimension of the output channel. The number of input feature maps of the network module in Figure 4 is 256, which are divided into four groups with 64 feature maps in each group. The idea of information exchange combination is the same as that in Figure 3.

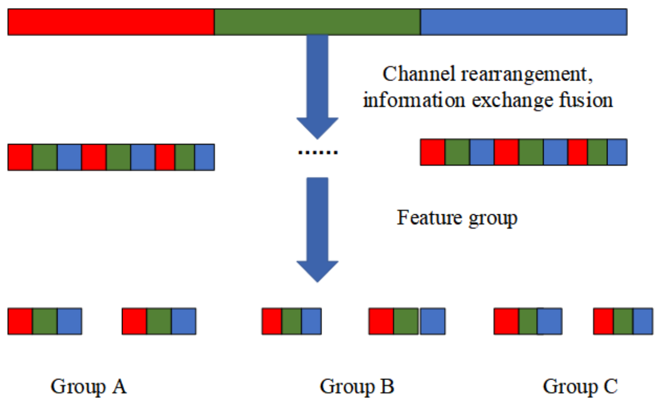

Fig. 5. Channel rearrangement

3.2. Down-sampling and channel amplification module

The traditional convolutional neural network usually adopts average pooling layer or maximum pooling layer for down-sampling, but this method ignores the importance and secondary of feature information, it does not consider the position information of the image, and regards the features of all positions as the same. For example, the receptive field information in the central area of an image is more complete and important than that in other areas, so different areas of an image correspond to different weights. In order to avoid the problem of decreasing accuracy due to the fuzzy effect of pooling layer, this paper uses 3×3 and 5×5 convolution kernels for down-sampling, so that the network learns the weights of different points by itself and combines them with the channel amplification process into a module. As shown in Figure 6, where the convolution step of 3×3 and 5×5 is 2, it is responsible for down-sampling. 1×1 (with step size 1) is responsible for raising and lowering dimensions of the channel.

3.3. Concentration and refining of feature information

After feature extraction of grouping module, the network can get rich feature information, but it is inevitable that there will be overlapping same information in these information. If

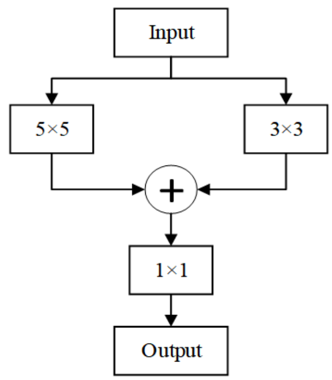

Fig. 6. Down-sampling module

all feature information is extracted and classified, the same information will be extracted repeatedly, resulting in a waste of computing resources. Therefore, a Feature Fusion and Concentration Convolution layer (FFCConv) is designed in this paper, which refines and condenses feature information by merging feature maps to solve the overlapping and redundant problems of feature information. The input of this layer is the feature map with $4 \times 4 \times 510$. Firstly, the feature information is rearranged through channels. After information exchange and fusion, the feature information is divided into A, B, C and D, and the number of feature maps in each group is 51, 102, 153 and 204, respectively. The specific process is shown in Figure 7. Convolution is carried out for each feature graph in Group A. For group B, two layers are combined into one for convolution; For group C, three layers are combined into one for convolution. For group D, four layers are convolved into one. The convolution mode is different from the conventional convolution mode, and the weighted mean convolution as shown in formula (1) is adopted. Where ω represents the weight of the corresponding position. Suppose that the weighted mean convolution calculation of the two feature graphs in group B is X_1 and X_2 respectively, then the calculation method of the feature combination result X_B in group B is shown in Formula (13). Similarly, the calculation method of the combined results of group C and group D is the same as that of group B. In this way, 51 1×1 feature maps are obtained for each group, and then 204 1×1 feature maps are obtained by integrating each group of channels. The number of information channels has been condensed and integrated from 510 to 204. In order to control the degree of information enrichment and prevent the loss of important feature information due to excessive information refining, this paper adopts the combination method as shown in Figure 7.

$$X_A = \frac{\omega_{11}x_{11} + \omega_{12}x_{12} + \cdots, \omega_{44}x_{44}}{16} \tag{10}$$

$$x_1 = \frac{\omega_{12}x_{12} + \omega_{13}x_{13} + \cdots, \omega_{44}x_{44}}{16} \times X_A \tag{11}$$

$$x_2 = \frac{\omega_{21}x_{21} + \omega_{22}x_{22} + \cdots, \omega_{44}x_{44}}{16} \times X_A \tag{12}$$

$$X_B = \frac{x_1 + x_2}{2} \tag{13}$$

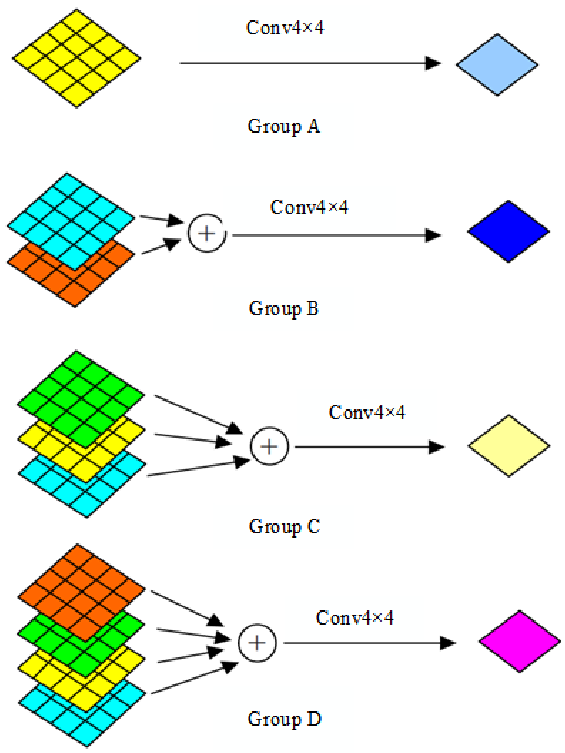

Fig. 7. Concentration and refining process

After many experiments, it is found that the network performance can be optimized by reusing Figure 4 module five times and Figure 5 module twice. The overall network configuration is shown in Table 1. Where c represents the number of channels and s represents the step size. The last down-sampling module in the network structure will keep the number of channels unchanged and only process the image size. Finally, the output of the network is fed into the full connection layer. Although the size of the image remains the same every time the network passes through a feature grouping extraction module, the image features extracted each time are different. These rich image features are helpful to improve the accuracy of image recognition.

$\omega_{11}x_{11}$	$\omega_{12}x_{12}$	$\omega_{13}x_{13}$	$\omega_{14}x_{14}$
$\omega_{21}x_{21}$	$\omega_{22}x_{22}$	$\omega_{23}x_{23}$	$\omega_{24}x_{24}$
$\omega_{31}x_{31}$	$\omega_{32}x_{32}$	$\omega_{33}x_{33}$	$\omega_{34}x_{34}$
$\omega_{41}x_{41}$	$\omega_{42}x_{42}$	$\omega_{43}x_{43}$	$\omega_{44}x_{44}$

Fig. 8. Feature map with 4×4 size

Table 1. Network configuration details

Input	Operator	c	s
$64 \times 64 \times 1$	$Conv3 \times 3$	64	1
$64 \times 64 \times 64$	Down-sampling module	129	2
$32 \times 32 \times 129$	[Network module 1]×5	129	1
$32 \times 32 \times 129$	Down-sampling module	256	2
$16 \times 16 \times 256$	[Network module 2]×2	256	1
$16 \times 16 \times 256$	Down-sampling module	510	2
$8 \times 8 \times 510$	Down-sampling module	510	2
$4 \times 4 \times 510$	FFCConv4×4	204	4
$1 \times 1 \times 204$	$Conv1 \times 1$	1024	1

3.4. New SqueezeNet Framework

As shown in Figure 9, the network structure of SqueezeNet is similar to that of the traditional convolutional neural network, CNN IS implemented by stacking the convolutional operation, but SqueezeNet is stacked by FireMoudle.

In this paper, the SqueezeNet model is improved in several parts: 1) The maximum pooling layer is added to the lower FireMoudle layer for fusion, and the over-fitting problem of small convolution kernel is improved. In this process, the size of the maximum pooling layer feature map and the fused FireMoudle feature map are guaranteed to match; 2) For FireMoudle layer feature map parameters, dynamic compression network surgery algorithm is used to dynamically link pruning and reduce network complexity; 3) Softmax with L2 norm constraint [25] is used to replace the original Softmax for classification and achieve better constraint effect through regularization. The model parameters are shown in Table 2. Pruning and splicing belong to dynamic network surgery process.

3.5. Dynamic Network Surgery

The commonly used model parameter pruning algorithm is to delete unimportant parameters by threshold value to compress the CNN model, but the importance of parameters often changes with the network performance, which leads to two common problems: 1) Important parameters may be deleted, which can reduce the accuracy of the model; 2) It takes a long time and the convergence is too slow. Dynamic network surgery (DNS) compression model adjusts the parameters [26]. The process of DNS contains pruning and

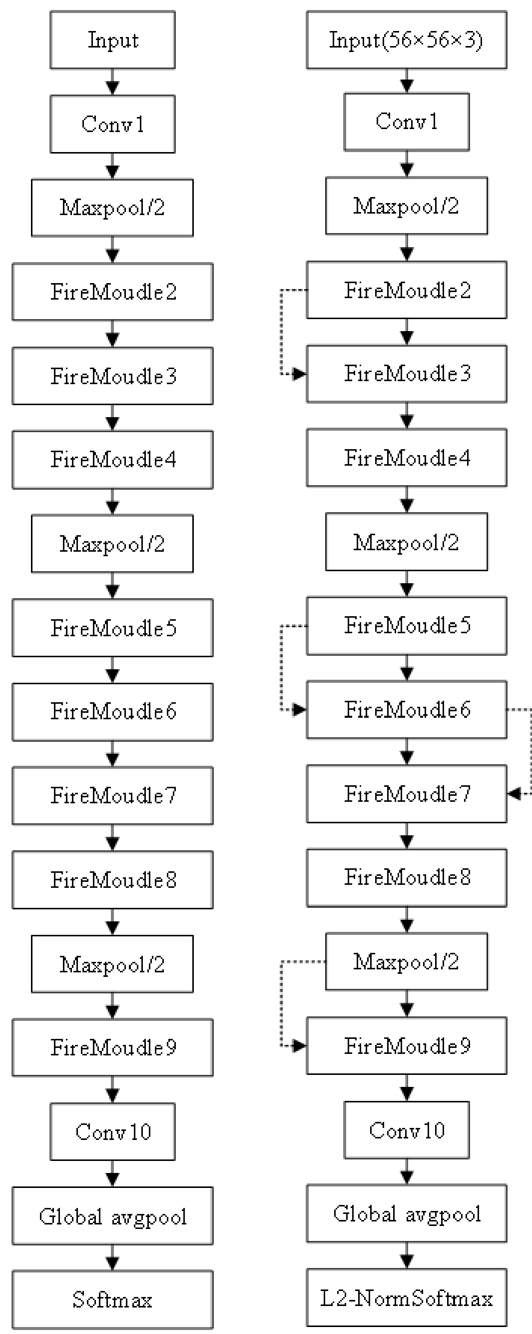

Fig. 9. Original and new SqueezeNet structure

Table 2. Improved SqueezeNet module parameters

Layer	Output size	step size	parameter	pruning	splicing
Input	$64 \times 64 \times 3$	none	none	none	none
Conv1	$64 \times 64 \times 96$	$1 \times 1 \times 96$	none	none	none
Maxpool1	$32 \times 32 \times 96$	$3 \times 3/2$	none	none	none
Fire2	$32 \times 32 \times 128$	none	11920	5746	7756
Fire3	$32 \times 32 \times 128$	none	12432	6258	8208
Fire4	$32 \times 32 \times 256$	none	45344	20646	25468
Maxpool4	$16 \times 16 \times 256$	$3 \times 3/2$	none	none	none
Fire5	$16 \times 16 \times 256$	none	49440	24742	32886
Fire6	$16 \times 16 \times 384$	none	104880	44700	63326
Fire7	$16 \times 16 \times 384$	none	111024	46236	73339
Fire8	$16 \times 16 \times 512$	none	188992	77581	10668
Maxpool8	$8 \times 8 \times 512$	$3 \times 3/2$	none	none	none
Fire9	$8 \times 8 \times 512$	none	197184	77581	10669
Conv10	$8 \times 8 \times 3755$	$1 \times 1 \times 3755$	513000	103400	24380
Avgpool10	$1 \times 1 \times 3755$	$8 \times 8 \times 1$	none	none	none

splicing, as shown in Figure 10. Training is synchronized with compression, which can reduce a large number of parameters while ensuring accuracy. Pruning is a compression network model. Splicing is to make up for the loss of precision caused by incorrect pruning and restore splicing of incorrect pruning. It not only improves the learning efficiency, but also better closes to the compression limit. For problem 2, there are two ways to accelerate the training speed: 1) reduce the deletion probability of parameters and improve the convergence speed; 2) separate FireMoudle and convolutional layer for parameter tailoring.

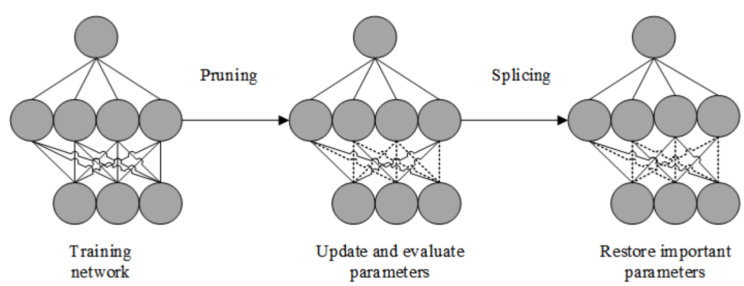

Fig. 10. Dynamic network surgery strategy

Formula (14) shows the loss function of the network.

$$\begin{cases} min_{W_k,T_k} L(W_k \odot T_k) \\ s.t. T_k^{(i,j)} = h_k(W_k^{(i,j)}), \forall (i,j) \in I \end{cases} \tag{14}$$

$L(W_k \odot T_k)$ is the network loss function. $\odot$ stands for matrix Hadamard product. $(h_k(w))$ is a classification function. If the judgment is important, it is 1; otherwise, it is 0. T_k is the 0-1 matrix, indicating the connection state of the network, whether it is pruned or not. I is a member of the matrix W_k.

Classification function $(h_k(w))$ is shown in Equation (15). The importance of parameters is based on the absolute value of weight, and two thresholds a_k, b_k are set, where $b_k = a_k + T_k$.

$$h_k(W_k^{(i,j)}) = \begin{cases} 1, b_k \leq |W_k^{(i,j)}| \\ T_k^{(i,j)}, a_k \leq |W_k^{(i,j)}| \prec b_k \\ 0, a_k > |W_k^{(i,j)}| \end{cases} \tag{15}$$

After W_k and T_k are determined, the value of W_k is updated by Formula (16). β is positive learning efficiency. Equation (16) not only updates important parameters, but also updates parameters that have been identified as unimportant or invalid to the loss reduction function, that is, the parameter set as 0 in T_k is still updated.

$$W_k^{(i,j)} \leftarrow W_k^{(i,j)} - \beta \frac{\partial L(W_k \odot T_k)}{\partial (W_k^{(i,j)} T_k^{(i,j)})}, \forall (i,j) \in I \tag{16}$$

Pruning and splicing in the algorithm is a iteration process, which is realized by constantly changing the values of the weight W_k and T_k of the connection until the iteration number $iter$ reaches a preset value. The dynamic network algorithm is shown in **Algorithm 1**.

Algorithm 1 Dynamic network surgery algorithm

Input: training set X, related model W_k, learning rate α, learning strategy f

```
Initializing W_k = W_k0, T_k = 1, 0 ≤ K ≤ C, β = 1, iter = 0
Select the network trained by X
Forward propagation and loss calculation are obtained from W_0 ⊙ T_0 to W_c ⊙ T_c
Back propagation yields model output and loss function gradient
for k = 0 to C do
    Update T_k with function h_k and existing W_k
    Update W_k through the formula and the existing loss function gradient
    X_3 = ProjectConv(X_2, groups = 2, noBN)
end for
Update iter = iter + 1, β = f(α, iter)
Until iter reaching the preset maximum value
Output: (W_k, T_k)
```

3.6. Network Optimization

The overall network has a Batch Normalization (BN) layer following the convolutional layer of each module. The distribution of input data is easy to change, and the change will be amplified with the deepening of the network. Therefore, in order to adapt to the change, the network model has to learn the new data distribution of the change, which leads to slower and slower training convergence. BN layer can prevent the change of data distribution through normalization, and enlarge the influence factor of input on loss function, so that the gradient of back propagation becomes larger. To apply the data after BN normalization and increase the gradient, it is necessary to increase the learning rate to accelerate the convergence speed.

Assuming that the input of layer l of the network is $Z^{[l-1]}$, it is normalized and the superscript $[l-1]$ is ignored, then the normalization process is shown in formulas (17) (19):

$$\mu = \frac{1}{m}\sum_{i=1}^{m} Z_i \tag{17}$$

$$\delta^2 = \frac{1}{m}\sum_{i=1}^{m} (Z_i - \mu)^2 \tag{18}$$

$$Z^i_{norm} = \frac{Z_i - \mu}{\sqrt{\delta^2 + \epsilon}} \tag{19}$$

Where, m is the number of samples contained in a single training data set. ϵ is constant, $\epsilon = 10^{-7}$. The input value z^i_{norm} has a mean of 0 and a variance of 1. However, if the data is forced to be normalized, the original feature learning of the network will be affected, resulting in the decline of the network expression ability. By introducing two adjustable parameters γ and β, formula (20) can be obtained, namely:

$$Z = \gamma \times Z^i_{norm} + \beta \tag{20}$$

In the formula, γ and β are learnable parameters, similar to weight and bias, they are obtained by gradient descent algorithm. The value of Z can be changed by adjusting the values of γ and β. If $\gamma = \sqrt{\delta^2 + \epsilon}$, $\beta = \mu$, then $Z = Z_i$. Therefore, the introduction of parameters γ and β can enable the network to obtain the distribution of features to be learned and enhance the expression ability of the network.

3.7. Fusion Process

The fusion of the maximum pooling layer and the lower FireMoudle layer not only improves the over-fitting problem of small convolution kernel, but the bottom feature has a higher resolution and contains more location and detail information. However, there is a lot of noise, while the top feature has a lower resolution but poor perception of details. The fusion of high-level features and low-level features can improve the detection effect of small targets (points in handwritten Chinese characters). The feature map learned by the front layer can be accessed by the back layer, and the whole network shares some features, making the model more compact.

The feature map obtained by pooling layer and the feature map obtained by FireMoudle are fused to get a new feature map. The process is shown in Equation (21).

$$\begin{cases} x(n) = f[w^i down(x)^j + b^j] f(w^i x^i + b^i) \\ s.t. i + j = n \end{cases} \tag{21}$$

Where n, i and j represent the number of new feature maps, the number of feature maps extracted by pooling layer and the number of feature maps processed by FireMoudle respectively.

3.8. L2-Softmax

For the given test input, Softmax estimates and normalizes the probability value of each category through the hypothesis function, and obtains the normalized probability value of the category, as shown in Equation (22). In pattern recognition tasks, multiple categories can be separated effectively and easily implemented. But there are also obvious disadvantages: 1) If there are too many categories, there will be matching problems; 2) Limited by the processing method of maximizing conditional probability, it is more suitable for high-quality images than difficult and rare images.

$$L = -\frac{1}{M} \sum_{i=1}^{M} log \frac{e^{\varphi}}{\sum_{j=1}^{C} e^{\varphi}} \tag{22}$$

where $\varphi = W_{y_i}^T f_i$.

Feature visualization is achieved when the output of the last hidden layer is restricted to 2. Figure 11 shows the feature distribution obtained by Softmax and L2-Softmax [27] on the mnist data set. The accuracy of L2-Softmax is higher than that of Softmax.

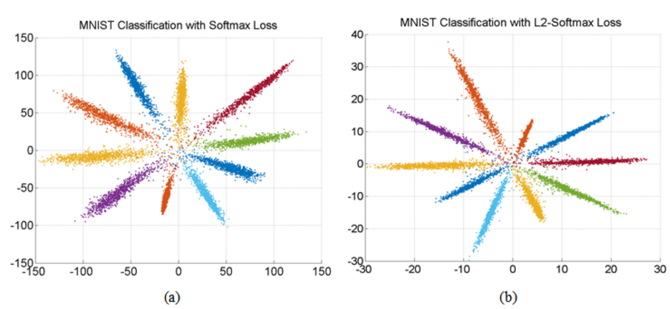

Fig. 11. Comparison of Softmax and L2-Softmax feature distributions in the mnist dataset

Since there are many categories of handwritten Chinese characters in this paper, Softmax constrained by L2-norm is adopted for classification. Based on the norm constraints, images of the same category are closer to each other in the normalized feature space, and images of different categories are farther apart. The attention given to sample averaging works well for poor quality samples.

Formula (23) is the normalization of L2-Softmax category probability values. Where $f(x_i)$ is an input image with size M. y_i represents the class description of the $i-th$ object, where only one element is 1. $f(x_i)$ is the D-dimensional feature description before the final fully connected layer. C is the number of categories. W and b represent trainable weights and deviations in the network respectively.

$$\begin{cases} min(L) \\ s.t.||f(x_i)||_2 = \alpha, \forall i = 1, 2, \cdots, M \end{cases} \tag{23}$$

The implementation of L2 constraint in the network is shown in Figure 12. Softmax process directly normalizes Softmax loss to obtain probability values, while L2-Softmax introduces L2 formatting layer and Scale layer before Softmax output. L2 formatting layer normalizes the input feature x into a unit vector. The Scale layer scales the unit vector to a fixed radius according to the given parameter a. Given that the value obtained from the simultaneous training of parameter a and other network parameters is too large, this paper directly fixes a as a small constant, which has a better effect.

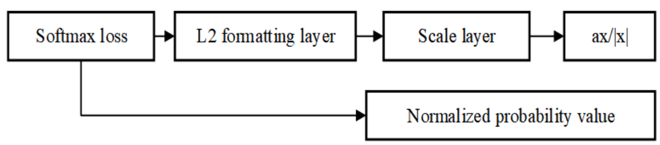

Fig. 12. L2-Softmax process

4. Experimental Results and Analysis

In this paper, CASIA-HWDB (V1.1) data set [28] and ICDAR-2013 data set [29] are selected to train and test the designed neural network model. The two data sets include 68645 first-level Chinese character samples. The collected data set is the original sample. In order to improve the performance of the neural network training model, it is necessary to perform data amplification and error selection processing on the training set. The training set plays an important role in the training of the model. In order to keep the original samples of the training set and improve the performance of the training model as much as possible, only the obvious errors are processed in this paper. At the same time, in order to train the test of the model and prevent over-fitting, the training set is only lightly processed, and the test set is not changed.

Due to the deepening of the network, there may be the risk of over-fitting. At this time, if the data set is small, it is easy to fit the characteristics of the data set. Therefore, data augmentation is introduced to protect against the risk of over-fitting. First, the original sample of the training set is randomly flipped up and down. The training set after processing has more than 500 sample images for each category of handwritten Chinese characters, which makes the training model be better.

The different handwriting depth of Chinese characters in the data set will affect the recognition accuracy, and the image contrast enhancement operation will be carried out.

$$D(x,y) = \frac{(I(x,y) - I_{min}) \times 255}{I_{max} - I_{min}} \tag{24}$$

In Formula (24), I_{max} and I_{min} are the maximum and minimum gray pixel values of the original image respectively. $I(x,y)$ is the pixel value of the original image. $D(x,y)$ is the pixel value of the target image.

If the image size is too large, the network burden will be increased, and if it is too small, the recognition performance will be reduced. The nearest neighbor interpolation method is used to normalize the image size to 56×56. The combination of gradient features can improve the effectiveness and accuracy of handwritten Chinese character recognition. The features of handwritten Chinese characters are extracted from the 8 directions of 0, $\pi/4$, $\pi/2$, $3\pi/4$, $3\pi/2$, $7\pi/4$, which can cover horizontal, vertical, skimming and curling strokes of Chinese characters. The horizontal and vertical gradients are obtained by Sobel operator, and the feature graphs of eight directions are obtained by parallelogram decomposition. Finally, the average gradient image is obtained by superposition.

In Figure 13, the upper left corner is one Chinese character of the original image, the middle is the image after image enhancement processing, the upper right corner is the average gradient image after gradient image superposition, and the following 8 images are gradient images in corresponding directions.

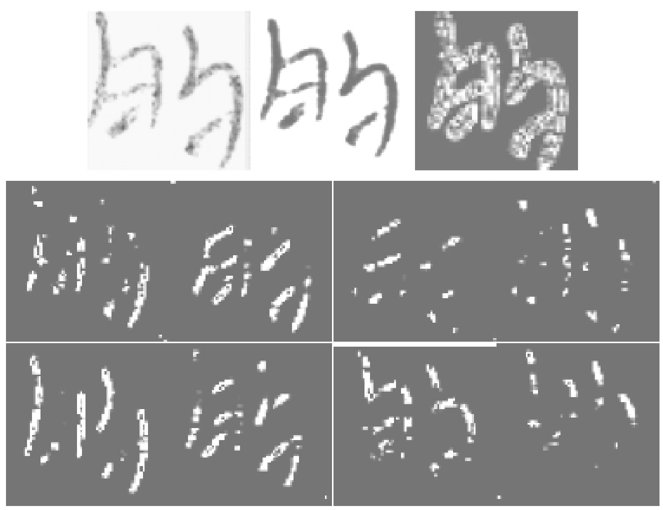

Fig. 13. Pretreatment of Chinese word

4.1. Experimental environment and implementation

All experiments in this paper used CUDA parallel computing architecture on Ubantu 18.04 system, and built PyTorch framework on the basis of CUDNN accelerated computing library [30]. The graphics card used in the experiment was NVIDIA GEForce

GTX3090 (24G) with 32.0GB of memory and Intel(R) Core(TM) i7-6950XCPU 3.00GHZ. We set network hyperparameters as shown in Table 3. FireMoudle sets the compression ratio to 0.5. The filter number of 3×3 accounts for 0.25 of the total filter number.

Table 3. Hyperparameters set

Parameter	Value
Loss function	L2-Softmax
Optimizer	Adadelta
Activation function	ReLU
Dropout	0.5
Iterations	20
Batch Size	128

4.2. Comparison analysis

The model has a total of 10^4 training iterations, and data is saved every 100 steps. Therefore, the loss and accuracy of model training and testing are shown in figures 14 17. In order to optimize the results of the training model, the whole experiment process is monitored and the parameters are adjusted timely according to the training situation. We set the learning rate to 0.1 to accelerate the convergence speed; When training step= 40000, the learning rate is reduced to 0.01 to stabilize the training; When training step=80000, the learning rate is reduced to 0.001. At the end of training, training loss and test loss converge stably to 0.48 and 0.17, respectively. The accuracy of training and testing is 0.9875 and 0.9716, respectively. Finally, top-5 experiments are carried out on the basis of the trained model, and the accuracy rate is as high as 99.37%.

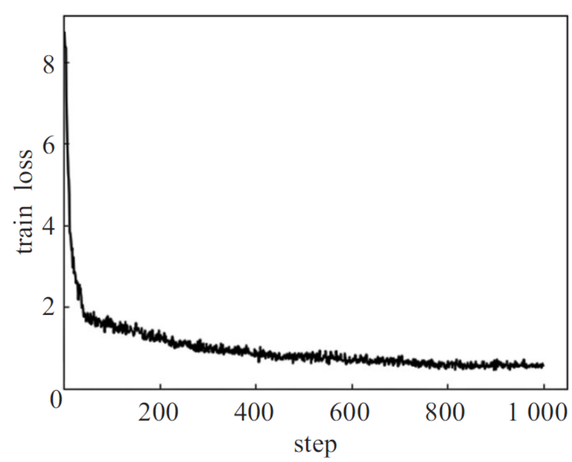

Fig. 14. Training loss

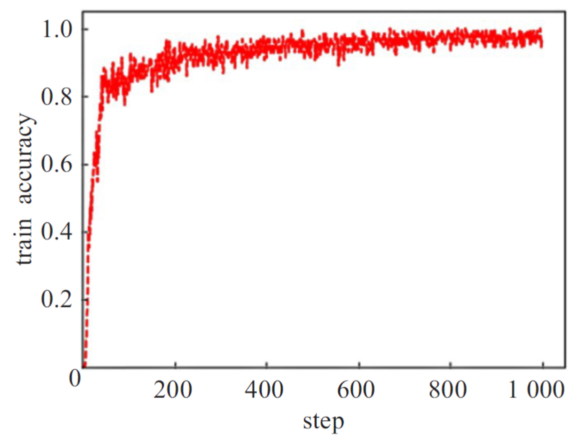

Fig. 15. Training accuracy

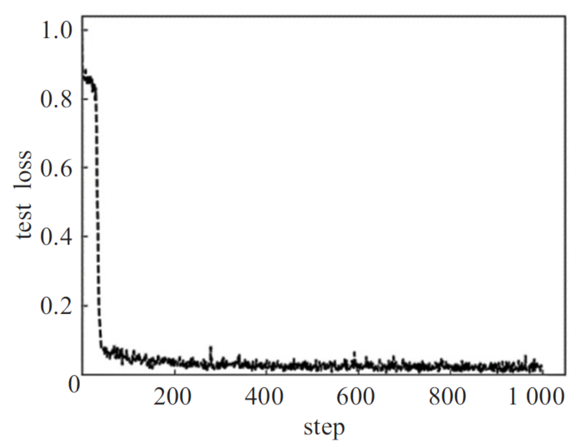

Fig. 16. Test loss

Table 4 shows the ablation results. The proposed method in this paper has higher accuracy in all tests than a single SqueezeNet model. For Top-1, Top-2 and Top-5, the accuracy values of proposed method are 98.65%, 99.72% and 99.84% respectively.

Table 5 shows the comparison results between the proposed method and other advanced methods including PageNet [31], SON [32], HHCR [33], ROA [34]. Compared with the original residual network, the accuracy of PageNet has obvious advantages, but it is much different from that of other advanced models. The SON model has only 5 convolutional layers, and the extraction of feature information is insufficient, which affects the classification effect. In HHCR, multiple neural networks are used for training, and then the average of these multiple results is taken. This method requires a lot of work

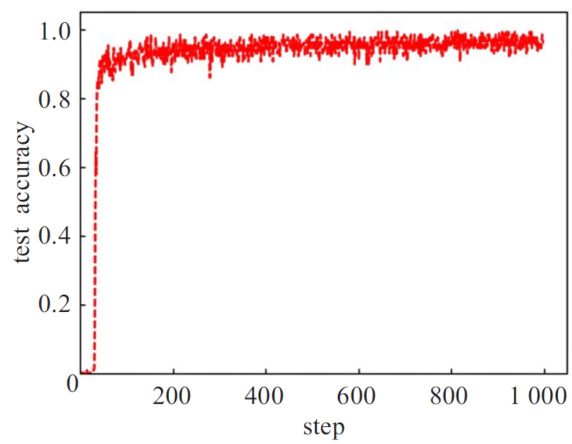

Fig. 17. Test accuracy

Table 4. Ablation result

Model	Top-1 accuracy %	Top-2 accuracy %	Top-5 accuracy %
SqueezeNet	95.43	98.18	97.25
Proposed	98.65	99.72	99.84

and time-consuming experiments. ROA improves the traditional convolution method and adopts the convolution method without shared weights, which achieves good results, but there is a problem of too large parameter volume. The results obtained by the new method are better than those obtained by other methods.

Table 5. Ablation result

Model	Top-1 accuracy %	Top-2 accuracy %	Top-5 accuracy %
PageNet	92.19	93.34	92.08
SON	93.45	92.67	93.84
HHCR	94.87	94.96	95.52
ROA	95.77	96.23	96.79
Proposed	98.65	99.72	99.84

5. Conclusions

Aiming at the problem of difficult recognition for handwritten Chinese characters, a multilevel stacked SqueezeNet model is proposed. By seeking an appropriate fusion comparison strategy and combining the advantages of SqueezeNet in handwritten Chinese character recognition, the multilevel stacked feature group extraction module is used to extract the deep abstract feature information of the image. Experimental results show that the proposed fusion model achieves better recognition results than other advanced machine learning methods on open data sets. The next research direction is to further improve the network structure and explore better fusion ways. And handwritten Chinese character recognition is often used in some embedded devices, its resources limit the application of this method, so how to compress the network structure, save resources and speed up the calculation is also the focus of the research direction.

Acknowledgments. This work was supported by "Project Name: Image Processing based handwritten Chinese character recognition and correction scoring method and application research; Project number: 22B520044".

References

1. Li, Y., et al.: "Fast and Robust Online Handwritten Chinese Character Recognition with Deep Spatial & Contextual Information Fusion Network," *IEEE Transactions on Multimedia*, doi: 10.1109/TMM.2022.3143324.
2. Hu, M., Qu, X., Huang, J., et al.: "An End-to-End Classifier Based on CNN for In-Air Handwritten-Chinese-Character Recognition," *Applied Sciences,* Vol. 12, No. 14, 6862. (2022)
3. Li, P., Laghari, A., Rashid, M., Gao, J., Gadekallu, T., Javed, A., Yin, S.: "A Deep Multimodal Adversarial Cycle-Consistent Network for Smart Enterprise System," *IEEE Transactions on Industrial Informatics,* Vol. 19, No. 1, 693-702. (2023). doi: 10.1109/TII.2022.3197201.
4. Dan, Y., Zhu, Z., Jin, W., et al.: "S-Swin Transformer: simplified Swin Transformer model for offline handwritten Chinese character recognition," *PeerJ Computer Science,* Vol. 8, e1093. (2022)
5. Dan, Y., Zhu, Z., Jin, W., et al. "PF-ViT: Parallel and Fast Vision Transformer for Offline Handwritten Chinese Character Recognition," *Computational Intelligence and Neuroscience,* Vol. 2022. (2022)
6. Peng, D., et al.: "Recognition of Handwritten Chinese Text by Segmentation: A Segment-annotation-free Approach," *IEEE Transactions on Multimedia,* doi: 10.1109/TMM.2022.3146771.
7. Wang, L., Yin, S., Hashem, Alyami., et al.: "A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images," *Geoscience Data Journal,* (2022). https://doi.org/10.1002/gdj3.162
8. Teng, L., Qiao, Y.: "BiSeNet-oriented context attention model for image semantic segmentation," *Computer Science and Information Systems,* Vol. 19, No. 3, pp. 1409-1426. (2022)
9. Zhang, K., Zuo, W., Chen, Y., et al.: "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing,* 2017, 26(7): 3142-3155.
10. Zhu, Y., Zhang, H., Huang, X., et al.: "Visual normalization of handwritten Chinese characters based on generative adversarial networks," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 36, No. 3, 2253002. (2022)
11. Hu, S., Wang, Q., Huang, K., et al.: "Retrieval-based language model adaptation for handwritten Chinese text recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, 1-11. (2022)
12. Yan, K., Guo, J., Zhou, W.: "A Novel Method for Offline Handwritten Chinese Character Recognition Under the Guidance of Print," *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11C14, 2021, Proceedings, Part II. Cham: Springer International Publishing*, 106-117. (2021)
13. Zhou, M., Zhang, X., Yin, F., et al.: "Discriminative quadratic feature learning for handwritten Chinese character recognition," *Pattern Recognition*, Vol. 49, 7-18. (2016)

14. Qu, X., Wang, W., Lu, K., et al.: ”Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network,” *Pattern recognition letters*, Vol. 111, 9-15. (2018)
15. Diao, H., Chen, C., Yuan, W., et al.: ”Deep residual networks for sleep posture recognition with unobtrusive miniature scale smart mat system,” *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 15, No. 1, 111-121. (2021)
16. Cui, R., Liu, H., Zhang C.: ”A deep neural framework for continuous sign language recognition by iterative training,” *IEEE Transactions on Multimedia*, Vol. 21, No. 7, 1880-1891. (2019)
17. Guo, Z., Zhou, Z., Liu, B., et al.: ”An Improved Neural Network Model Based on Inception-v3 for Oracle Bone Inscription Character Recognition,” *Scientific Programming*, Vol. 2022. (2022)
18. Heo, Y.: ”Loss function optimization for cnn-based fingerprint anti-spoofing,” *International Journal of Computer and Information Engineering*, Vol. 15, No. 6, 344-348. (2021)
19. Wang, Z., Du, J.: ”Fast writer adaptation with style extractor network for handwritten text recognition,” *Neural Networks*, Vol. 147, 42-52. (2022)
20. Wu, Y., Yin, F., Liu, C.: ”Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models,” *Pattern Recognition*, Vol. 65, 251-264. (2017)
21. Greco, A., Saggese, A., Vento, M., et al.: ”Gender recognition in the wild: a robustness evaluation over corrupted images,” *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, 10461-10472. (2021)
22. McGuire, M., Moore, T.: ”Prediction of tornado days in the United States with deep convolutional neural networks,” *Computers & Geosciences*, Vol. 159, 104990. (2022)
23. Kohler, M., Langer, S.: ”On the rate of convergence of fully connected deep neural network regression estimates,” *The Annals of Statistics*, Vol. 49, No. 4, 2231-2249. (2021)
24. Baaran, E.: ”Classification of white blood cells with SVM by selecting SqueezeNet and LIME properties by mRMR method,” *Signal, Image and Video Processing*, Vol. 16, No. 7, 1821-1829. (2022)
25. Yang, M., Tjuawinata, I., and Lam, K.: ”K-Means Clustering With Local d?-Privacy for Privacy-Preserving Data Analysis,” *IEEE Transactions on Information Forensics and Security*, Vol. 17, 2524-2537. (2022) doi: 10.1109/TIFS.2022.3189532.
26. Guo, Y., Yao, A., Chen, Y.: ”Dynamic network surgery for efficient dnns,” *Advances in neural information processing systems*, Vol. 29. (2016)
27. Sun, L., Li, W., Ning, X., et al.: ”Gradient-enhanced softmax for face recognition,” *IEICE TRANSACTIONS on Information and Systems*, Vol. 103, No. 5, 1185-1189. (2020)
28. Wu, Y., Yin, F., Liu, C.: ”Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models,” *Pattern Recognition*, Vol. 65, 251-264. (2017)
29. Karatzas, D., Shafait, F., Uchida, S., et al.: ”ICDAR 2013 robust reading competition,” *2013 12th international conference on document analysis and recognition. IEEE*, 1484-1493. (2013)
30. Yuan, Y., Xu, Z., and Lu, G.: ”SPEDCCNN: Spatial Pyramid-Oriented Encoder-Decoder Cascade Convolution Neural Network for Crop Disease Leaf Segmentation,” *IEEE Access*, Vol. 9, 14849-14866.(2021) doi: 10.1109/ACCESS.2021.3052769.
31. Peng, D., Jin, L., Liu, Y., et al.: ”PageNet: Towards End-to-End Weakly Supervised Page-Level Handwritten Chinese Text Recognition,” *International Journal of Computer Vision*, Vol. 130, No. 11, 2623-2645. (2022)
32. Xu, X., Yang, C., Wang, L., et al.: ”A sophisticated offline network developed for recognizing handwritten Chinese character efficiently,” *Computers and Electrical Engineering*, Vol. 100, 107857. (2022)
33. Huang, G., Luo, X., Wang, S., et al.: ”Hippocampus-heuristic character recognition network for zero-shot learning in Chinese character recognition,” *Pattern Recognition*, Vol. 130, 108818. (2022)

34. Wang, Y., Yang, Y., Ding, W., et al.: ”A residual-attention offline handwritten Chinese text recognition based on fully convolutional neural networks,” *IEEE Access*, Vol. 9, 132301-132310. (2021)

Yuankun Du is with the School of Big Data and Artificial Intelligence, Zhengzhou University of Science and Technology, Associate Professor, research direction: Artificial Intelligence, Big data analysis.

Fengping Liu (Senior Engineer) is with the School of Information Engineering, Zhengzhou University of Science and Technology, Research interests: Software development, Information management and information system.

Zhilong Liu (Lecturer) is with the Library of Henan Institute of Animal Husbandry Economics, Research interests: computer network, information system.

Received: December 10, 2022; Accepted: April 28, 2023.