IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2021, accepted January 5, 2022, date of publication January 14, 2022, date of current version January 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3143795

SwordNet: Chinese Character Font Style Recognition Network

XUDONG LI, JINGYI WANG, HAIYANG ZHANG, YONGKE HUANG, AND HUIHUI HUANG
College of Software, Nankai University, Tianjin 300071, China

Corresponding author: Jingyi Wang (wangjing1@mail.nankai.edu.cn)

This work was supported in part by the National Science and Technology Major Project, China, under Grant 2018YFB0204304 and Grant 2021YFB0300104; in part by the Tianjin Natural Science Foundation under Grant 16JCYBJC15800; and in part by the Fundamental Research Funds Nankai University for the Central Universities under Grant z1a2085588.

ABSTRACT Chinese characters have been created into many font styles, such as official script, running script, and regular script. Other than that, some famous calligraphers, such as Ouyang Xun and Yan Zhenqing, have produced fonts with their style. Being able to detect and recognize these font styles quickly and accurately has essential applications in graphic design, page layout, handwriting identification, and other use cases. Distinguishing between font styles requires professional knowledge, which almost inevitably leads to errors for unprofessional people. Therefore, this paper presents a sword-like model based on a convolutional neural network with a sword structure to recognize font styles for Chinese characters. This model includes 15 convolutional layers. For each layer, we gradually increase the number of convolutional kernels to better extract the classification features of the input image. This paper uses four downsampling layers in the model. For each downsampling operation, the length and width of the image become half of their original values while the number of channels gradually increases, leading to a sword-like shape. As a result, we name our model as SwordNet. We also created a Chinese font dataset called the Nankai Chinese Font Style dataset and made it available on Github. Using the above dataset, we compared the accuracy of our model with six other state-of-the-art network models. The experiments showed that SwordNet could achieve an average recognition accuracy of 99.03% in multiple experiments, while the other six models can only achieve accuracy up to 94.91%. So we can conclude that SwordNet could perform better in font style recognition than other models.

INDEX TERMS Chinese character, convolution neural network, font style recognition, skip connection.

I. INTRODUCTION

During the long history of Chinese characters, a diverse set of font styles have been developed, such as Cursive Script, Clerical Script, and Small Seal Script. In addition to the evolution of strokes, Chinese characters have developed various font styles under the writing of calligraphers. There are more than 6000 types of Chinese characters, and there are great differences between different font styles of the same Chinese character. Font style recognition is essentially a classification task, but one of the difficulties lies in annotating font styles, especially those related to ancient Chinese characters. Typically, professional knowledge of written Chinese characters is required to distinguish font styles.

The associate editor coordinating the review of this manuscript and approving it for publication was Taehong Kim.

The font style that can recognize Chinese characters is significant to text-related work such as text recognition, artistic font style design, and handwriting identification [1]. Multiple font styles are used for readability and aesthetics considerations when formatting articles. For example, chapter titles and body text usually use different font styles [2]. We can reveal a certain amount of information by identifying different font styles in the article. There are many illegal and criminal cases where the criminal tried to mimic the victim's handwriting. For such cases, we usually need to call in human experts to judge on whether the handwriting is from a particular person or whether a criminal faked it. This not only consumes time and money but also inevitably leads to errors in some cases. If the handwriting could be automatically identified by a neural network model, it would not only save costs but also improve the accuracy of recognization. It is thus important to

design and build neural network models which can provide high accuracy.

To classify the font style of a Chinese character in an image, one first needs to extract features from that image, and then the features are used to train the model or for classification. At a high level, there are two approaches to extract features [3]–[5]. The first is a machine learning approach with manual involvement in extracting features, and the second is a deep learning approach without manual involvement in extracting features. Manual participation in feature extraction requires a lot of ad-hoc experiments to identify the most effective features for classification, with many of them are artificially constructed. Speeded up robust features (SURF) [6] and scale-invariant feature transform (SIFT) [7] are two examples. As we all know, the selection of the features has a great impact on recognition accuracy. As a result, this method has poor generalizability. On the other hand, deep learning methods have much better generalizability: the model automatically extracts the features, and it does not require human involvement in the feature extraction process.

Due to the advantage of automatical and more robust feature extraction in the deep learning approach, we also decided to use that approach in SwordNet, based on convolutional neural network (CNN). Given an image containing a Chinese character, SwordNet can recognize the font style of Chinese characters in the image end-to-end without complicated data pre-processing or human intervention. To test the generalizability of SwordNet, we collected ten common font styles and six ancient Chinese calligraphic works, including Shuowen Xiaozhuan, with a total of 18 different font styles. The experimental results showed that the recognition accuracy of our proposed model could be as high as 99.03%.

The structure of our proposed model SwordNet resembles a sword, including a convolutional layer containing 15 layers of convolution kernel size 3×3, using Global Average Pooling downsampling, and adding three skip connections to enhance the generalization ability of the model.

We make the following contributions in this paper.

1) We propose SwordNet, a new model for end-to-end Chinese font style recognition.
2) We release a new large-scale dataset, Nankai Chinese Font Style dataset, with 18 Chinese font styles covering standard printing font styles and works of ancient Chinese calligraphers. Each font style contains about 1000 images.
3) We evaluated our model together with six other state-of-the-art models and demonstrated that SwordNet could indeed achieve higher accuracy than others.

The remainder of this paper is organized as follows. Section II presents related work. In Section III, we describe the font dataset used in this paper. The detailed SwordNet model is presented in Section IV, and Section V shows the experimental results. Then, we conclude the paper.

II. RELATED RESEARCH

Before deep learning was invented, people had started to use machine learning for font style recognization. However, they typically required a manual feature selection process. This manual feature selection process was usually done in an ad-hoc fashion and the features picked were less robust, limiting the generalizability of a model. Zhu *et al.* [8] used a multi-channel Gabor filter to extract t-texture features to recognize font styles. Ding *et al.* [9] first did wavelet transformation to extract the wavelet features and then used Box-Cox transformation and Linear Discriminant Analysis (LDA) to get the style features. Finally, Modified Quadratic Distance Function (MQDF) was used for classification. Tao *et al.* [10] proposed the Sparse Discriminative Information Preservation (SDIP) and introduced the Local Binary Patterns (LBP) [11] descriptor to estimate the geometric structure of Chinese characters. They demonstrated that their scheme is much faster than schemes based on using wavelet features, and the best average recognition rate they achieved on 25 Chinese font style categories can be as high as 93.0%. Bennour [12] used Support Vector Machine (SVM) to recognize font styles for handwritten English. He first extracted Harris corner points from handwritten English character images and then used LBP for classification, which uses both local and global features. He achieved recognition accuracy rates up to 98.22%. Guo [13] proposed the Linear Discriminant Analysis Cauchy Estimator (LDACE) algorithm, which combines linear discriminant analysis and Cauchy estimator theory to extract the font style characteristics of Chinese characters. Experiments showed that LDACE achieves recognition accuracy of about 98% for 12 font data sets.

While the above methods were designed for font style recognition of independent Chinese characters, some character symbols, such as Arabic, are written using a large number of coherent strokes. This makes the segmentation task much more challenging. To handle this problem, Slimane *et al.* [14] proposed the sliding window based approach in which a sliding window was used to move over the image, removing the need for character segmentation. They then used extracted features for predication using the Gaussian Mixture Model (GMM) for font style classification. As we can see from these approaches, all of them have to perform complex preprocessing of the image to derive the classification features. There is no guarantee that features derived from a particular dataset would work equally well for other or future datasets.

With recent developments in deep learning, convolutional neural networks have shown superior performance in feature extraction and greatly surpass machine learning algorithms in some cases [15], [16]. Wang *et al.* [4] used a patch-based CNN model [17] to extract feature vectors of Chinese character images, and they achieved 97.53% recognization accuracy. Tao *et al.* [3] treated Chinese character font style recognition as a sequence of classification problems.

They combined a two-dimensional Long Short Term Memory Model (2DLSTM) with principal components to obtain the stroke trajectories of Chinese characters. The evaluation showed that their approach could achieve recognition rates as high as 97.77% while also demonstrating greater flexibility and robustness. In Lee and Ding [18], they used autoencoders that extract features during the training process according to the loss function, such as mean square error. They compared their method with traditional machine learning methods such as K-NN and demonstrated that their approach could deliver better recognization performance (with a recognization accuracy of 98.5% vs. 84.9%). In [19], Tang observed that the skeleton information of Chinese characters could be an important classification index. He proposed Skeleton Kernel, which uses a long narrow rectangular sliding window to extract skeleton features. For VGG19 [20] network, they showed that the recognition accuracy could be improved by about 10% with Skeleton Kernel. In Style and Content Supervision (SCS) network [21], it stacked two separate fully connected layer branches. These two branches extract the font style and content features of Chinese characters, respectively. Then these two features are mixed using a bilinear model and fed into a softmax layer for classification. Experimental results showed that SCS achieved recognition accuracy of 88.06% on a Chinese data set containing 91 fonts. Deep learning-based approaches remove the need for complex pre-processing steps and the manual feature construction process. Instead, the models can automatically learn the features based on the training dataset, ensuring that the most appropriate features are used.

Given the advantage of the deep learning-based approach for this problem, we also looked into leveraging neural networks in this work. Compared to other deep learning approaches, SwordNet is the first to use skip connection and Global Average Pooling (GAP), to improve the robustness of the model. Experiment results show that SwordNet can achieve better recognition accuracy on datasets with different sizes than existing deep learning approaches and recognize written English characters.

While the above methods demonstrated good recognition results, none of them publicly available their datasets. This results in two problems. Firstly, it becomes challenging to reproduce their results without access to their dataset. Secondly, we don't have a representative dataset that includes many categories with different font styles. One of the larger Chinese character datasets is HCL2000 [22] is a handwritten Chinese dataset consisting of 3755 first-level simplified Chinese characters. It was written by 1,000 participants of different ages and occupations, with various educational backgrounds. THU-HCD [23] is another dataset published by a group from Tsinghua University. Similar to HCL2000, THU-HCD also only contains the first-level simplified Chinese character samples. However, the dataset is much larger and is divided into ten subsets based on how well the handwritings were organized. It also has more participants: about 2000 people contributed to this dataset. While the above two datasets were primarily used for offline handwritten Chinese characters recognition, CASIA-HWDB [24] has been used for both offline and online handwritten Chinese characters recognition. Besides, the offline handwritten Chinese characters are further divided into independent characters or texts. CASIA-HWDB is more often used for Chinese character recognition.

In contrast, our Nankai Chinese Font Style (NCFS) dataset contains both handwriting and standard printing. Furthermore, the Chinese characters in each font style include not only the first-level simplified Chinese characters but also some rare characters and ancient Chinese characters that cannot be represented by Unicode encoding.

III. NANKAI CHINESE FONT STYLE DATASET

In this section, we described the details of our dataset: Nankai Chinese Font Style (NCFS) dataset [25]. It contains three parts.

1) Ancient Chinese calligraphic characters: we selected six calligraphic works written by five authors in different styles. The text is an essay named ''Thousand Characters Classic''. It is a popular enlightenment reading written by Zhou Xingsi in ancient China about one thousand years ago and has been translated into other languages, such as English and French. The following authors wrote the selected six calligraphic works: Han Lishu, Liu Gongquan, Mi Fu in running script, Ouyang Xun, and Yan Zhenqing (on the Duobao Tower Stele and the Qin Li Stele). The calligraphic works were first scanned and then segmented so that each image contained only a single character. Next, each image is denoised and binarized. Fig. 1 shows some of these characters, written by Mi Fu and Ouyang Xun.
2) Standard computer font characters: we selected ten common True Type Font (TTF) fonts and used them to generate the characters for the same essay: ''Thousand Characters Classic''. TTF is the font standard used by the Windows operating system. Fig. 2 shows the selected fonts. SimHei, SimKai, and SimLi are commonly used daily, while CAIYUN, HUPO, STK, and YTK add certain artistical features. XC, XK, and XS are font styles with unique Chinese features. Considering some rare characters that cannot be displayed, we generated 987 character images for each TTF font. Fig. 3 shows some of the images contained in YTK and HUPO font styles in the converted NCFS dataset.
3) Ancient Chinese Symbols: for the last part, we got some ancient Chinese glyph symbols not defined by Unicode from the Zhonghua Book Company Song Ti 15 Plane font library.[1] Specifically, this font library includes two types of symbols: ''Ancient Characters'' and 'Shuo Wen Xiao Zhuan''. Some examples are shown in Fig. 4.

[1] http://www.ancientbooks.cn/

FIGURE 1. The first part of the NCFS dataset obtained from the ancient Chinese calligraphy work "Thousand Characters Classic". (a) is Mi Fu's Running Script, and (b) is written by Ouyang Xun.

FIGURE 2. Ten common fonts TTF file, The font styles in the left column are CAIYUN, SimHei, HUPO, SimKai, and SimLi. The font styles in the right column are STK, XC, XK, XS, and YTK.

We store each image using the PNG format. The images in each category are randomly split into the training set and the test set in the ratio of 8:2. Table 1 provides a summary for the NCFS dataset: it contains a total of 18 font styles, and the number of images contained in the training set and test set for each font style is also shown.

TABLE 1. Nankai Chinese font style dataset.

Font Name	Training Set	Test Set	Total
Han Lishu	800 images	200 images	1000 images
Liu Gongquan	800 images	200 images	1000 images
Mi Fu's Running Script	800 images	200 images	1000 images
Ouyang Xun	800 images	200 images	1000 images
Yan Zhenqing's Duobao Tower Stele	800 images	200 images	1000 images
Yan Zhenqing's Qin Li Stele	800 images	200 images	1000 images
CAIYUN	790 images	197 images	987 images
SimHei	790 images	197 images	987 images
HUPO	790 images	197 images	987 images
SimKai	790 images	197 images	987 images
SimLi	790 images	197 images	987 images
STK	790 images	197 images	987 images
XC	790 images	197 images	987 images
XK	790 images	197 images	987 images
XS	790 images	197 images	987 images
YTK	790 images	197 images	987 images
Ancient Chinese Characters	580 images	145 images	725 images
Shuo Wen Xiao Zhuan	800 images	200 images	1000 images

IV. METHOD DESCRIPTION

A. PROPOSED ARCHITECTURE

Fig. 5 shows the structure of SwordNet, which has a shape like a sword. It includes 15 convolutional layers, each with a convolutional kernel size of 3×3. The stride parameter is 2, and the padding parameter is "same". Stacking more convolutional layers can achieve better accuracy. But as the number of layers increases, the feature values extracted from each layer will be distributed in the saturation interval of the activation function, and then the gradient disappears. To handle the above issue, we used Batch Normalization (BN) [26] for each convolutional layer to accelerate the convergence and improve the accuracy of the network. The BN-layer can make the feature values satisfy the distribution law with mean 0 and variance 1. As a result, the activation function becomes more sensitive to the eigenvalues and produces a larger gradient to accelerate convergence.

Convolution and Pooling are both linear calculation processes. In order to introduce nonlinear factors, an activation function layer needs to be added after the BN-layer. Many commonly used activation functions in deep convolutional neural networks, such as the Sigmoid activation function. It maps the input to between 0 and 1. The disadvantage of the Sigmoid activation function is that the larger the input value, the smaller the gradient value. In this case, it is easy for the gradient to disappear in the deep network. Rectified Linear Unit (ReLU) is also a commonly used activation function. The ReLU activation function maps inputs less than 0 to 0, and inputs greater than or equal to 0 to the input value itself. When the network model reversely updates the parameters, many derivation operations are required. The ReLU activation function is adopted by many models because of its convenience in derivation [5], [27], [28]. We use the ReLU activation function after BN-layer in SwordNet.

To simplify the model and increase the computational speed, we use the MaxPooling layer in SwordNet. The function of the MaxPooling layer is to downsample

FIGURE 3. Images in png format are generated by converting common TTF font files from the NCFS dataset's second part. (a) is YTK, (b) is HUPO.

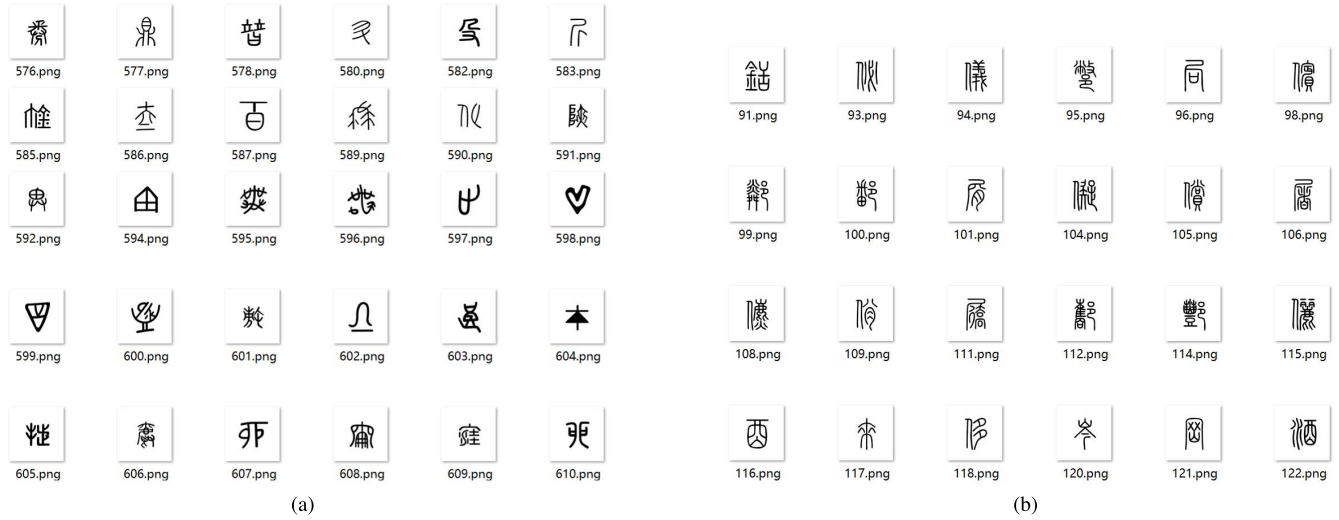


FIGURE 4. Images in png format generated by converting ancient Chinese glyph symbol TTF font files form the third part of the NCFS dataset. (a) is Ancient Chinese Characters, and (b) is Shuo Wen Xiao Zhuan.

the image. We set the MaxPooling layer after the 2nd, 5th, 10th, and 14th ReLu activation function layers, and the sampling step is set to 2. Therefore, after each MaxPooling layer, the length and width of the image become half of the original. The number of channels of the image increases with convolution kernels. During the whole convolution process, the length and width of the image gradually decrease, and the number of channels gradually increases. The entire structure looks like a sword, leading to the name for our neural network: SwordNet.

B. SKIP CONNECTION

The ResNet model [29] and SH Tsang's research[2] showed that adding skip connection to a convolutional neural network has the effect of updating the model layers. The skip connection changes the distribution of the model parameters, making them more uniform, improving model accuracy. Therefore, we were inspired to add three skip connections to SwordNet, as shown by the solid black connection line in Fig. 5. These three skip connections add the output of the 3rd, 8th, and 12th layers as additional input for the following three layers.

C. GLOBAL AVERAGE POOLING

The convolution operation extracts the features of the image, and the fully connected layer is used for classification. The fully connected layer receives the flattened feature values as input, resulting in processing a large number of parameters. To handle this challenge, Lin *et al.* [30] proposed Global Average Pooling (GAP), which takes the average value of each feature value into the softmax layer to reduce the number of parameters in the fully connected layer. We used GAP in the penultimate layer of SwordNet, as shown in the purple

[2]Review: U-Net+ResNet–The Importance of Long & Short Skip Connections (Biomedical Image Segmentation). Visit the web page at https://medium.com/datadriveninvestor/review-u-net-resnet-the-importance-of-long-short-skip-connections-biomedical-image-ccbf8061ff43

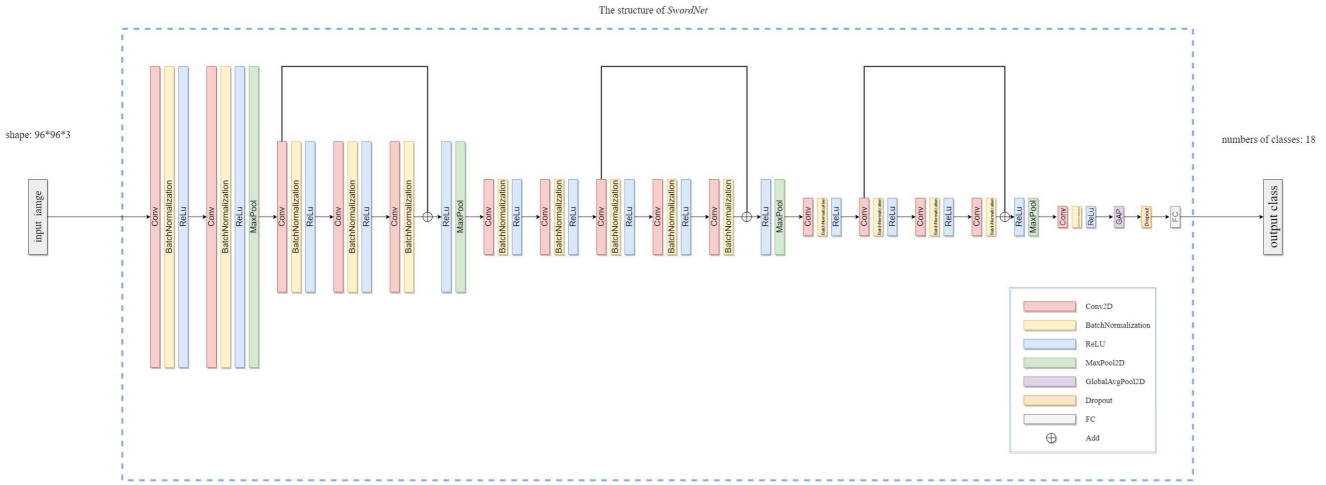


FIGURE 5. SwordNet structure diagram. The input image is in format 96 × 96 × 3 and passes through 15 convolutional blocks in sequence. Each consisting of a convolutional layer, a BN layer, a ReLU activation function layer, and a MaxPooling layer. SwordNet also includes three skip connections, adding GAP at the end, and SwordNet will output one of the predicted 18 font style types.

box in Fig. 5. SwordNet also used Dropout, which randomly deactivates some neurons to reduce overfitting. Table 2 shows the structure of each layer of SwordNet and the shape of the output image, where Conv_Block is a convolution block, which is composed of convolution layer, batch normalization, and ReLU activation function layer.

V. EXPERIMENTS

A. EXPERIMENTAL DETAILS

The data used for training are RGB images of 96 × 96 pixels. We used 96 pixels because 96 × 96 pixels is sufficient to distinguish the different Chinese characters clearly, and many other studies [31]–[33] also used this size. Our model can process images of various sizes. We set the number of channels to 3 because these are RGB images.

The number of epochs used in training also impacts the model accuracy. Too many epochs can easily cause overfitting, and too few can easily make the training parameters less than optimal. We trained the model for 30 epochs in the experiment, which showed a good balance. We set the batch_size to 16 : 16 images are processed as one batch simultaneously. Suppose the input data used for training form a particular order or pattern. In that case, it is easy to fall into the local optimum when calculating the gradient in the training process. This results in poor generalization of the model. To avoid the image order on the training results, we randomly took out one batch_size of images at a time.

We used Small batch stochastic gradient descent (SGD) as the optimization function and set the initial learning rate to 0.1. To speed up the training process and obtain better results, we used a learning rate scheduler to automatically adjust the size of the learning rate according to the current fitting effect and loss. We adopted the categorical_crossentropy loss function to evaluate the gap between the predicted and true values of the model during the training process, and the parameters are updated based on this gap during backpropagation. The categorical_crossentropy loss function is defined as follows:

$$loss = -\frac{1}{n}\sum_{i} y_i \ln \hat{y}_i. \tag{1}$$

In (1), y denotes the actual label, $\hat{y}$ denotes the predicted output, and n denotes the total number of samples.

TABLE 2. The structure of SwordNet model.

Layer Name	Layer	Output Shape
Input	96 × 96 RGB image	(96,96,3)
Conv_Block × 2	3 × 3 Conv2D, 64. BN, ReLU	(96,96,64)
	3 × 3 Conv2D, 128. BN, ReLU	(96,96,128)
MaxPool	MaxPool2D, stride 2	(48,48,128)
Conv_Block × 3	3 × 3 Conv2D, 128. BN, ReLU	(48,48,128)
	3 × 3 Conv2D, 128. BN, ReLU	(48,48,128)
	3 × 3 Conv2D, 128. BN, ReLU	(48,48,128)
MaxPool	MaxPool2D, stride 2	(24,24,128)
Conv_Block × 5	3 × 3 Conv2D, 256. BN, ReLU	(24,24,256)
	3 × 3 Conv2D, 256. BN, ReLU	(24,24,256)
	3 × 3 Conv2D, 256. BN, ReLU	(24,24,256)
	3 × 3 Conv2D, 256. BN, ReLU	(24,24,256)
	3 × 3 Conv2D, 256. BN, ReLU	(24,24,256)
MaxPool	MaxPool2D, stride 2	(12,12,256)
Conv_Block × 4	3 × 3 Conv2D, 512. BN, ReLU	(12,12,512)
	3 × 3 Conv2D, 512. BN, ReLU	(12,12,512)
	3 × 3 Conv2D, 512. BN, ReLU	(12,12,512)
	3 × 3 Conv2D, 512. BN, ReLU	(12,12,512)
MaxPool	MaxPool2D, stride 2	(6,6,512)
Conv_Block × 1	3 × 3 Conv2D, 1024. BN, ReLU	(6,6,1024)
GAP	-	(1024)
Dropout	0.5	(1024)
Output	18 Softmax	18

The structure of each layer of the SwordNet model and the size of the feature map. When the output shape is a triplet, it means that the output feature map is a multi-dimensional matrix, where the first element represents the height of the feature map, the second element represents the width of the feature map, and the third element represents the number of channels of the feature map; When output shape is a tuple, it means that the output is a vector, where the elements represent the dimensions of the vector.

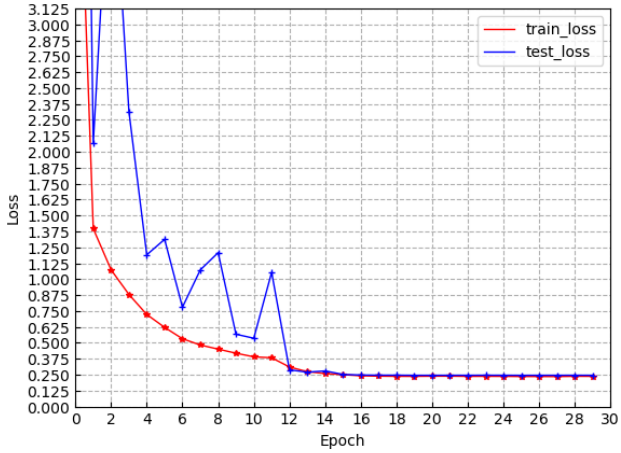


FIGURE 6. The loss variation of SwordNet on the training and test sets during the training process. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained; The vertical axis represents the loss. The red line segment represents the loss change on the training set, and the blue line segment represents the loss change on the test set.

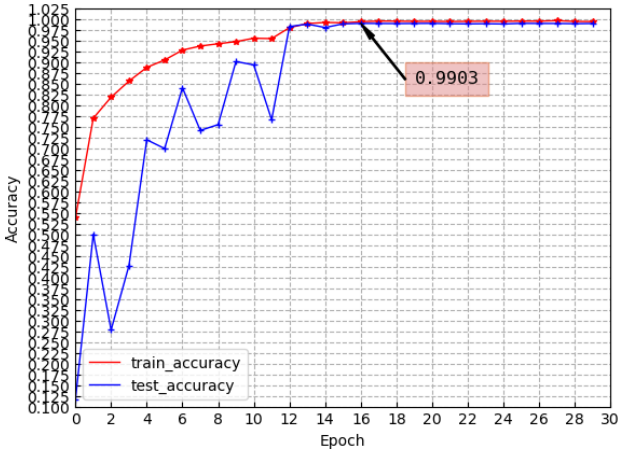


FIGURE 7. The change of accuracy of SwordNet on the training and test sets during training. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained; The vertical axis represents the accuracy rate. The red line represents the change in accuracy on the training set, and the blue line represents the change in accuracy on the test set.

B. COMPARISON WITH OTHER NETWORK MODELS

The experimental results show that the recognition accuracy of our proposed model can be as high as 99.03%. Fig. 6 shows the variation of the loss of SwordNet with epoch on the training and test sets during training, and Fig. 7 shows the variation of the accuracy with epoch. We can see that the loss tends to converge from the 16th round and the highest recognition accuracy of 99.03% occurs.

After training, SwordNet can load the weight parameters for font style recognition. Input an image containing only one Chinese character, SwordNet could predict the font style of this Chinese character. Fig. 8 shows six examples, the Chinese characters in the images input to SwordNet are from left to right, from top to bottom, they belong to CAIYUN, HUPO, Ancient Chinese Characters, Shuo Wen Xiao Zhuan, Mi Fu's Running Script, and Yan Zhenqing's Duobao Tower Stele. "class" indicates the font style predicted by SwordNet, and "prob" indicates SwordNet predicts the probability that a character belongs to a certain font style. We can see that SwordNet predicted all six examples correctly.

TABLE 3. Model comparison results.

Model	Total_Params	Test_Accuracy	Ref.
EfficientNet	4,072,629	0.8989	[36]
ShuffleNet	1,288,234	0.9071	[37]
AlexNet	6,229,714	0.9097	[34]
GoogleNet	5,992,002	0.9392	[20]
Vgg16Net	28,387,154	0.9457	[20]
ResNet	23,598,034	0.9491	[29]
SwordNet	16,186,322	**0.9903**	**Ours**

SwordNet has the highest accuracy.

During training, CNN uses gradient descent to update the weight parameters of the network based on the loss between the predicted and true values to obtain the most favorable features for classification. However, the features eventually extracted by CNN are often not interpretable, and these features are not as good as those extracted using traditional machine learning algorithms. To visualize the convolution process, an image containing a Chinese character is input, and Fig. 9 shows the input layer of SwordNet and the first 12 outputs of the middle five convolution layers. We can see that as the number of layers deepens, the features extracted by the CNN become more and more abstract, and the feature extracted from the 7th convolutional layer is no longer recognizable by a human. The feature extracted from the 11th convolutional layer resembles pixel blocks, which are much different from the predefined features and are not interpretable.

Using the same dataset NCFS, we compared the accuracy of our model with six other state-of-the-art network models. Table 3 shows the number of parameters of each model and the recognition accuracy results on the test set. Under the same experimental conditions, SwordNet obtained the highest accuracy of 99.03%, followed by ResNet [29] with an accuracy of 94.91%. AlexNet [34], GoogleNet [35] and Vgg16Net [20] also obtained better recognition accuracy, and EfficientNet [36] had the lowest accuracy which was only 89.89%. SwordNet's parameters are 16,186,322, which is less than one-half of that of Vgg16Net and ResNet, but the accuracy is improved by about 4% compared with these two models. We concluded that SwordNet could perform better in font recognition than other models.

Fig. 10 shows the accuracy of the testing set when we varied the number of epochs for training. SwordNet reached the highest accuracy with 16 epochs of training, while for others, their accuracy kept improving, even with more than 20 epochs of training. Fig. 11 and Fig. 12 shows the loss on the training set and the test set for each model when varying the number of epochs used in training. Both figures show that SwordNet can reach convergence with fewer epochs in training, requiring a shorter training time.

FIGURE 8. Input an image containing a Chinese character, and SwordNet predicts its font style. (a) is CAIYUN, (b) is HUPO, (c) is Ancient Chinese Characters, (d) is Shuo Wen Xiao Zhuan, (e) is Mi Fu's Running Script, and (f) is Yan Zhenqing's Duobao Tower Stele.

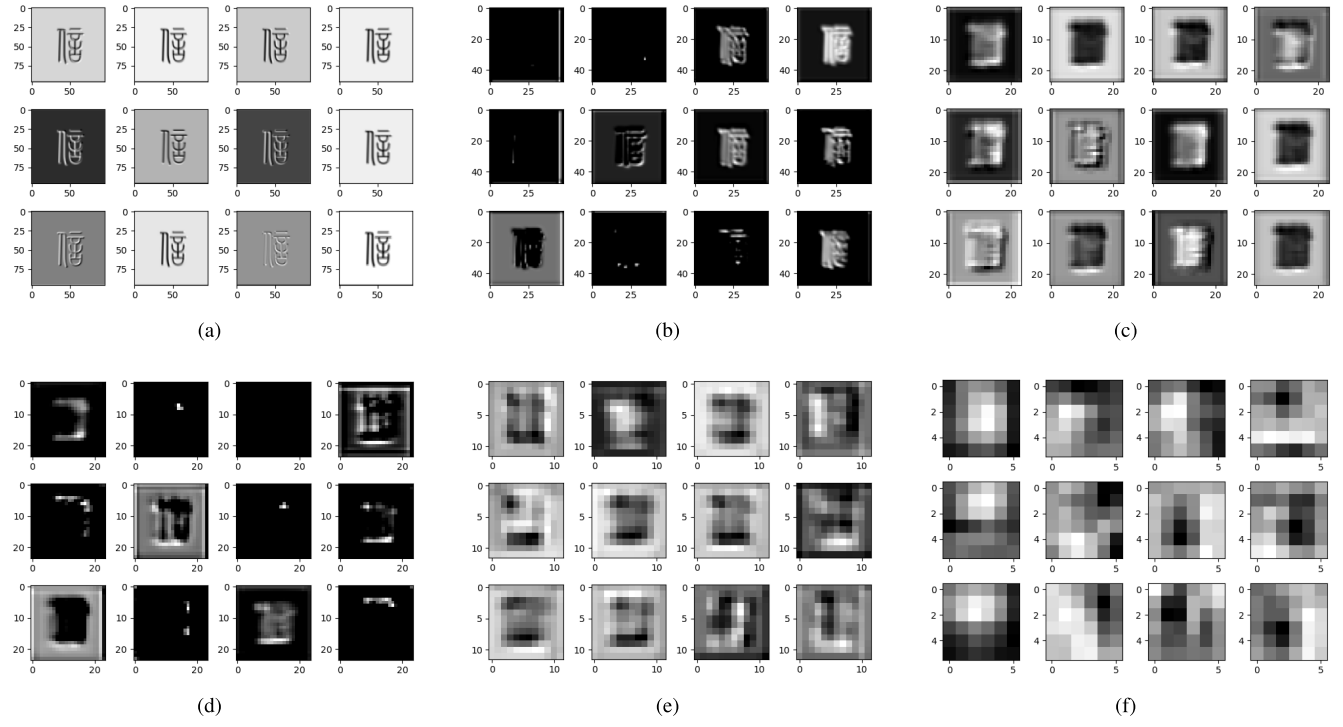


FIGURE 9. SwordNet intermediate convolutional layer feature map visualization. (a) is the output feature map of the first convolutional layer, (b) is the output feature map of the 5th convolutional layer, (c) is the output feature map of the 7th convolutional layer, (d) is the output feature map of the 11th convolutional layer, (e) is the output feature map of the 12th convolutional layer, and (f) is the output feature map of the 15th convolutional layer.

C. INFLUENCE OF THE SIZE OF DATASETS ON FONT STYLE RECOGNITION ACCURACY

With more data used for training, a model can learn and construct more useful features for prediction. To evaluate the effect of dataset size on model effectiveness, we varied the dataset size used in training. Except for the size of the datasets, all the experimental conditions were the same as before. Table 4 shows the experimental results. Dataset-1

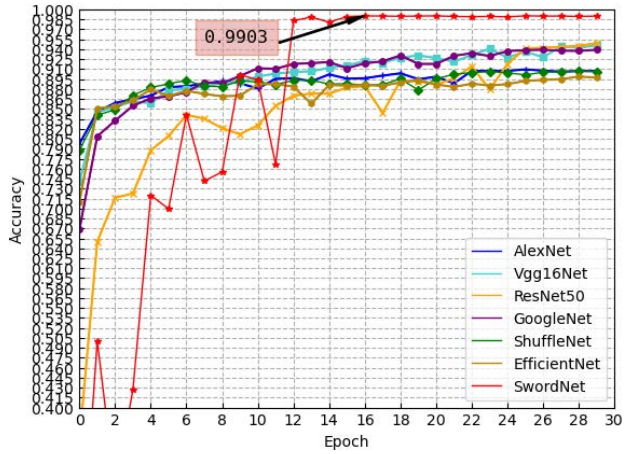


FIGURE 10. The variation of recognition accuracy of each model on the test set. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained. The vertical axis represents the accuracy rate. The red line represents the change of recognition accuracy of SwordNet proposed in this paper, and its highest accuracy is 0.9903.

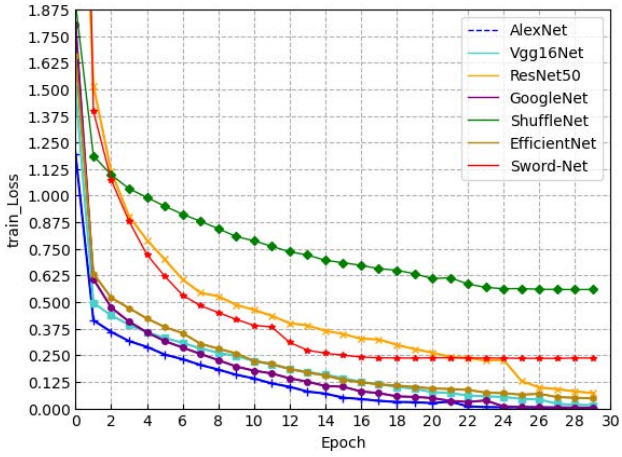


FIGURE 11. The loss variation of each model on the training set during the training process. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained. The vertical axis represents the loss. The red line segment represents the SwordNet model proposed in this paper, which has converged in the 16th round and converged earlier than the other models.

represents the initial dataset, and each font style category contains about 1000 images. Dataset-0.7 means that we only used 70% of the complete dataset for training and the same for other cases. We can see from the table that the recognition accuracy drops when reducing the size of the dataset. When we only used half of the images for training, SwordNet can still achieve 98.34% recognition accuracy, which is less than 1% lower when using the entire dataset. When we use only 20% or 10% of the dataset for training, SwordNet can still achieve recognition accuracy of 95% and 92.9% respectively, comparable with what other models can achieve with the entire dataset. With SwordNet, if we want to reduce the training time, we can reduce the dataset size by half while still achieving the recognition accuracy of 98%.

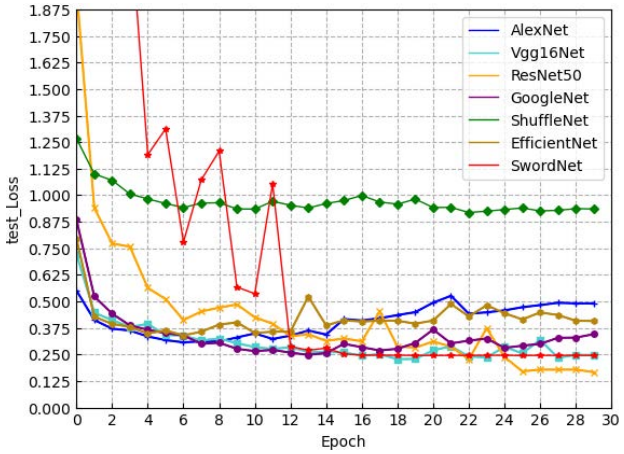


FIGURE 12. The loss variation of each model on the test set during the training process. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained; The vertical axis represents the loss. The red line segment represents the SwordNet model proposed in this paper.

TABLE 4. Comparison of the influence of dataset size on font style recognition accuracy.

DataSet	Test_Accuracy
DataSet-1	0.9903
DataSet-0.7	0.9882
DataSet-0.5	0.9834
DataSet-0.2	0.9574
DataSet-0.1	0.9290

Dataset-1 represents the initial datasets, and each font style category contains about 1000 pictures. 0.7, 0.5, 0.2, and 0.1 represent 0.7, 0.5, 0.2, and 0.1 times the size of the initial datasets, respectively.

D. FONT STYLE RECOGNITION FOR OTHER LANGUAGES

To demonstrate the excellent applicability of SwordNet for other language scripts, we applied the proposed SwordNet to an English font dataset. The English font dataset is obtained by converting TTF font files into png format images. Twenty-one English font styles are selected, and each category contains 26 lowercase letters and 26 uppercase letters, i.e., each class includes 52 images containing only one letter. Each image is 96 × 96 pixels, and the number of channels is three (RGB images). These 52 images were then randomly divided into a training set and a test set in the ratio of 8:2 for SwordNet training.

We set the initial learning rate to 0.1, and the learning rate scheduler automatically adjusts the learning rate to speed up the training speed according to the learning effect of the model. The size of batch_size parameter is set to 4 because this dataset only contains 52 images in total. A smaller batch_size for a small sample dataset allows the model to achieve a better recognition accuracy.

After 30 epochs of iterative learning, SwordNet achieved recognition accuracy of 96.23%. Fig. 13 shows the recognition accuracy when we varied the number of epochs for training. We achieved the highest recognition accuracy of 0.9623 using 17 epochs in training. The entire training process takes 1 to 2 minutes. We demonstrated that SwordNet can not only recognize various font styles and font styles

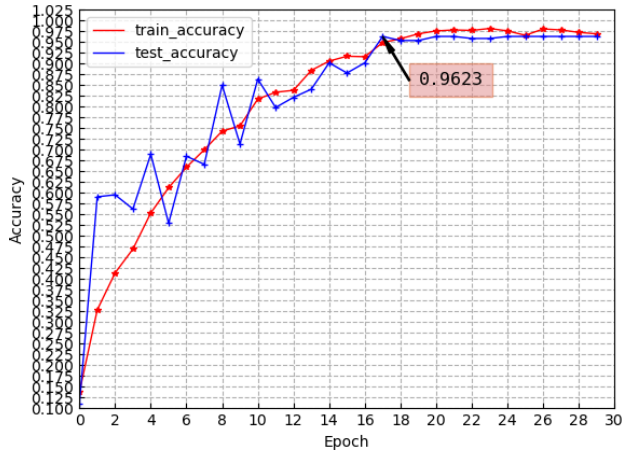


FIGURE 13. The variation of recognition accuracy of SwordNet on the training and test sets of the English font dataset during the training process. The horizontal axis represents the number of epochs used for training. And a total of 30 epochs were iteratively trained; the vertical axis represents the accuracy rate. The red line represents the change in accuracy on the training set, and the blue line represents the change in accuracy on the test set.

written by different calligraphers for Chinese characters but also be applied to recognize font styles for other languages. Section V-C shows that if we can get a larger dataset for training, SwordNet can continue to improve the recognition performance.

VI. CONCLUSION

In this paper, we propose SwordNet, a font style recognition model for Chinese characters with a sword structure. SwordNet uses MaxPooling and Global Average Pooling to downsample, and addes three skip connections in the middle layer to enhance the model's generalization ability. SwordNet obtained 99.03% accuracy on the 18 class NCFS dataset. The experimental results show that SwordNet achieved the highest recognition accuracy compared with the other six CNN models, such as ResNet, ShuffleNet, or GoogleNet. This paper also explores the effect of reducing the size of the dataset on the recognition accuracy of the model and concludes that SwordNet can still obtain more than 98% accuracy when reducing the size of the dataset to half of the initial size. The proposed model can recognize ancient Chinese font styles such as ancient Chinese characters and Shuo Wen Xiao Zhuan, which would be helpful for future studies related to ancient Chinese books. SwordNet has no limitation on the number of font styles and can be expanded to include new font styles in the future. It can also be trained using personal handwriting datasets for handwriting identification. Besides, SwordNet can also recognize font styles for English and other languages, making it a general technique useful for font style recognition.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable comments, and also thank Dr. Xing Lin for his help in improving the writing of this paper.

REFERENCES

[1] X. Bai, B. Shi, C. Zhang, X. Cai, and L. Qi, "Text/non-text image classification in the wild with convolutional neural networks," *Pattern Recognit.*, vol. 66, pp. 437–446, Jun. 2017, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320316303922

[2] N. Amara and S. Gazzah, "Une approche d'identification des fontes arabes," in *Proc. Conférence Internationale Francophone I'Ecrit Document*, 2004, pp. 1–7.

[3] D. Tao, X. Lin, L. Jin, and X. Li, "Principal component 2-D long short-term memory for font recognition on single Chinese characters," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 756–765, Mar. 2016.

[4] Y. Wang, Z. Lian, Y. Tang, and J. Xiao, "Font recognition in natural images via transfer learning," in *MultiMedia Modeling*, K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O'Connor, Y.-S. Ho, M. Gabbouj, and A. Elgammal, Eds. Cham, Switzerland: Springer, 2018, pp. 229–240.

[5] Y. Chang, "Chinese font recognition based on convolution neural network," in *Proc. 3rd Int. Conf. Automat., Mech. Control Comput. Eng. (AMCCE)*. Paris, France: Atlantis Press, 2018, pp. 562–566, doi: 10.2991/amcce-18.2018.97.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.[Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314207001555

[7] M. Moradi and P. Abolmaesumi, "Medical image registration based on distinctive image features from scale-invariant (SIFT) key-points," in *Proc. Int. Congr. Ser.*, vol. 1281, 2005, p. 1292, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0531513105002530

[8] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1192–1200, Oct. 2001.

[9] X. Ding, L. Chen, and T. Wu, "Character independent font recognition on a single Chinese character," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 195–204, Feb. 2007.

[10] D. Tao, L. Jin, S. Zhang, Z. Yang, and Y. Wang, "Sparse discriminative information preservation for Chinese character font categorization," *Neurocomputing*, vol. 129, pp. 159–167, Apr. 2014, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231213009740

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[12] A. Bennour, "Automatic handwriting analysis for writer identification and verification," in *Proc. 7th Int. Conf. Softw. Eng. New Technol.* New York, NY, USA: ACM, Dec. 2018, pp. 1–7, doi: 10.1145/3330089.3330129.

[13] Y. Guo, "Linear discriminant analysis Cauchy estimator for single Chinese character font recognition," *J. Phys., Conf. Ser.*, vol. 1069, Aug. 2018, Art. no. 012177, doi: 10.1088/1742-6596/1069/1/012177.

[14] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, "A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution," *Pattern Recognit. Lett.*, vol. 34, no. 2, pp. 209–218, Jan. 2013.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision-ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 818–833.

[17] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, "DeepFont: Identify your font from an image," in *Proc. 23rd ACM Int. Conf. Multimedia*. New York, NY, USA: ACM, Oct. 2015, pp. 451–459, doi: 10.1145/2733373.2806219.

[18] C.-C. Lee and J.-J. Ding, "Automatic Chinese handwriting verification algorithm using deep neural networks," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2019, pp. 1–2.

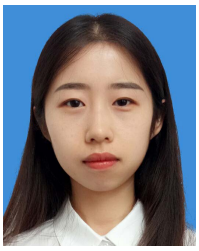
[19] W. Tang, Y. Su, X. Li, D. Zha, W. Jiang, N. Gao, and J. Xiang, "CNN-based Chinese character recognition with skeleton feature," in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 461–472.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[21] W. Tang, Y. Jiang, N. Gao, J. Xiang, Y. Su, and X. Li, "SCS: Style and content supervision network for character recognition with unseen font style," in *Neural Information Processing*, T. Gedeon, K. W. Wong, and M. Lee, Eds. Cham, Switzerland: Springer, 2019, pp. 20–31.

[22] H. Zhang, J. Guo, G. Chen, and C. Li, "HCL2000—A large-scale handwritten Chinese character database for handwritten character recognition," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, Jul. 2009, pp. 286–290.
[23] Q. Fu, X. Ding, T. Li, and C. Liu, "An effective and practical classifier fusion strategy for improving handwritten character recognition," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1038–1042.
[24] *CASIA Online and Offline Chinese Handwriting Databases*. Accessed: 2021. [Online]. Available: http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html
[25] *Nankai Chinese Font Style Dataset*. Accessed: 2021. [Online]. Available: https://github.com/JingY1W/Nankai-Chinese-Font-Style-Dataset.git
[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37. Jul. 2015, pp. 448–456.
[27] S. Huang, Z. Zhong, L. Jin, S. Zhang, and H. Wang, "DropRegion training of inception font network for high-performance Chinese font recognition," *Pattern Recognit.*, vol. 77, pp. 395–411, May 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317304235
[28] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognit.*, vol. 70, pp. 163–176, Oct. 2017, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302017
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[30] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014, *arXiv:1312.4400*.
[31] P. Melnyk, Z. You, and K. Li, "A high-performance CNN method for offline handwritten Chinese character recognition and visualization," *Soft Comput.*, vol. 24, no. 11, pp. 7977–7987, May 2019, doi: 10.1007/s00500-019-04083-3.
[32] Q. Xu, X. Bai, and W. Liu, "Multiple comparative attention network for offline handwritten Chinese character recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 595–600.
[33] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, "Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition," *Pattern Recognit.*, vol. 72, pp. 72–81, Dec. 2017, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302558
[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[36] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Jun. 2020, pp. 6105–6114.
[37] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

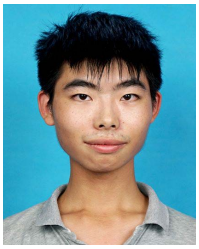
XUDONG LI received the B.S. degree from the Department of Computer and Systems Science, Nankai University, China, in 1997, and the Ph.D. degree from the College of Information Science and Technology, Nankai University, in 2003. He is now working with the College of Software, Nankai University. He held a number of patents for his many innovations. His research interests include software architecture, neural networks, operating systems, and distributed computing.



JINGYI WANG was born in Hebei, China, in 1996. She is currently pursuing the master's degree in software engineering with the College of Software, Nankai University, Tianjin, China. Her main research interests include deep neural networks and text recognition.

HAIYANG ZHANG is currently pursuing the master's degree with the College of Software, Nankai University, Tianjin, China. His main research interests include deep learning and Chinese character recognition.

YONGKE HUANG was born in Hebei, China, in 1998. He received the B.S. degree in software engineering from Nankai University, in 2019. His main research interests include blockchain and distributed storage.



HUIHUI HUANG is currently pursuing the master's degree with the College of Software, Nankai University, Tianjin, China. His main research interests include deep learning and website development.