Research Article

Feature Extraction and Identification of Calligraphy Style Based on Dual Channel Convolution Network

Ziyan Wang

Xi’an Jiaotong University, Xi’an 710049, Shaanxi, China

Correspondence should be addressed to Ziyan Wang; wzyabbamm525@stu.xjtu.edu.cn

Received 25 February 2022; Revised 17 March 2022; Accepted 28 March 2022; Published 16 May 2022

Academic Editor: Chin-Ling Chen

Copyright @ 2022 Ziyan Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To improve the effect of calligraphy style feature extraction and identification, this study proposes a calligraphy style feature extraction and identification technology based on two-channel convolutional neural network and constructs an intelligent calligraphy style feature extraction and identification system. Moreover, this paper improves the C3D network model and retains 2 fully connected layers. In addition, by extracting the outline skeleton and stroke features of calligraphy characters, this paper calculates the feature weight and authenticity determination function and constructs an authenticity identification system. The experimental study shows that the calligraphy style feature extraction and identification system based on the dual-channel convolutional neural network proposed in this paper has a good performance in calligraphy style feature extraction and identification.

1. Introduction

The digitized historical calligraphy works, books, and signatures contain tens of thousands of individual calligraphy characters, covering the original calligraphy works or the photocopies of various historical periods and famous calligraphers. These calligraphic works are displayed in the CADAL portal. Users can search these calligraphic works based on metadata information such as the title of the work and the name of the calligrapher. At the same time, some calligraphy characters can also be retrieved by content-based methods by inputting or drawing a calligraphy character image. However, for digital calligraphy works with a large amount of data, the original outline-based calligraphy character retrieval technology has gradually been unable to meet the needs of user applications because it takes too long. At the same time, in the actual service, the calligraphy character database needs to be continuously supplemented. When adding new calligraphy characters to the calligraphy character database, in order to obtain complete metadata, each calligraphy character needs to be manually identified and entered into the database by the administrator. This is a very time-consuming and labor-intensive job. In addition, with the continuous increase of digitized calligraphy works, users also put forward new demands for better use of these calligraphy characters. There are many users who wish to use the words of a famous calligrapher in these calligraphy resources to inscribe their study or company. However, among those characters specified by the user, there may be some characters that are not present in the existing works of the individual specified calligraphers. There is a need for a technique that can generate such calligraphic characters.

When identifying calligraphy samples through CNN, we must first extract the features of the image. The particularity of calligraphy samples is reflected in two aspects: the writing background is simple, usually black and white. The various writing styles are not completely unique, but there is a mutual reference. Moreover, there are many similar properties in the features. Before the calligraphy font is recognized, a good feature extractor is needed to extract the features of the samples [1]. In the Caffe neural network framework, the LMDB data storage format is used to store images, so it is necessary to format the training samples and test samples. The convolutional neural network is mainly composed of three parts: a feature extraction layer, a fully connected layer, and a classifier [2]. Among them, the
feature extraction layer performs feature extraction on the input samples, and the obtained feature images are subjected to multiple visual enlargement processing to retain the representative image feature information, and then the features are vectorized through the fully connected layer. Finally, the retained eigenvectors are probability normalized by the classifier, and the categories are divided according to the principle of maximum probability [3].

This paper proposes a calligraphy style feature extraction and identification technology based on two-channel convolutional neural network, constructs an intelligent calligraphy style feature extraction and identification system, and improves the effect of calligraphy style feature extraction and identification.

2. Related Work

Reference [4] proposed a calligraphic character retrieval method based on the similarity of calligraphic character outlines. This method was used to test 20 calligraphic characters, and the precision was close to 90% under the 90% recall rate. Reference [5] proposed a method for retrieving calligraphy characters based on the skeleton features of calligraphy characters. Literature [6] proposes a fast multilayer retrieval method for calligraphy characters, which improves the retrieval speed through two-layer calligraphy character retrieval. The full-rate and precision rate remain basically unchanged. Reference [7] proposes a SC.HoG feature to describe calligraphy characters, which expresses the position information of a certain contour point of calligraphy characters and the distribution information of contour points around the contour point, so as to perform shape-based retrieval for calligraphy characters. Reference [8] first performs pruning processing, and filters out calligraphic character images that are impossible to be similar to the calligraphic character image to be retrieved by comparing the complexity of calligraphy characters, stroke density, and other characteristics. Dimensional calligraphy image feature data is used for retrieval, and PK-tree is used to improve retrieval speed. Reference [9] proposed an ancient book content retrieval method based on visual similarity. The above-mentioned content-based calligraphy character retrieval research shows that the extracted calligraphy characters have high feature dimensions. Therefore, with the continuous increase of calligraphy characters in the database, in the retrieval of large data calligraphy characters, when comparing and calculating the similarity, it still consumes a lot of time, and the retrieval speed becomes slower and slower, which cannot meet the needs of online services.

Reference [10] proposes a high-dimensional calligraphic character index method based on hybrid distance-tree (HD-Tree) to speed up retrieval. Given a query calligraphic character image, the hybrid distance tree index is used to complete high-dimensional calligraphy characters, image query. Reference [11] proposed an interactive high-dimensional vector indexing method based on partial-distance-map (PDM). PDM mainly establishes the relationship between the concept of Chinese character semantic level and the underlying shape features through user feedback to improve the retrieval speed of the massive calligraphy character database.

Literature [12] proposed a calligraphy character recognition method, which is based on the calligraphy character retrieval. First, the calligraphy character skeleton similarity retrieval method is used to retrieve the calligraphy in the database that is similar to the calligraphy character image to be recognized. Then, according to the semantic annotation of the retrieved similar calligraphic character images in the database, the recognition results are given. In the experiment, 300 calligraphy characters were tested and the recognition rate reached 96.3%. This method is based on retrieval, and the recognition time depends on the time required to retrieve similar calligraphic characters in the database. When the database is large, the time efficiency of this recognition method is very low. The polygon approximation method is to use polygon line segments to approximate the shape edge, and the minimum error is generally used to measure the quality of the approximation method used. Reference [13] proposed a shape polygon approximation method based on Hopfield neural network. Reference [14] proposed a relaxed iterative matching algorithm. This method uses the start and the end coordinates, length, and direction to describe stroke outline segments. A spline refers to a piecewise-defined polynomial parametric curve. For a given node vector, all splines of degree $n$ form a vector space. A basis for this space is a B-spline of degree $n$. Reference [15] proposed a B-spline curve matching method with affine invariance. The commonly used scale spaces are: Gaussian scale space, wavelet scale space, and shape scale space. The salient feature points of the target refer to those feature points that still exist in the simplified representation by tracking the positions of the feature points at different scales to give a simplified form of the shape.

The detection and description of image local features can be effectively used to identify objects. SIFT features are based on the principle that the key points of the local appearance on the object are independent of the size and rotation of the image. In addition, the tolerance of SIFT features to changes in light, noise, and viewing angle is also higher [16]. Based on the above advantages, the feature information of the image is relatively easy to extract. After the image feature database is established, it is easy to identify objects in the feature database, and the recognition rate is high. SIFT feature description also has a high detection rate for objects with partial occlusion, and only a small amount of SIFT image features can be used to calculate the position and orientation of key points. The SIFT method has high operating efficiency, and the SIFT feature library has a large amount of information, and the recognition speed is close to real-time operation, so the algorithm is suitable for efficient and accurate matching of massive data [17].

The speeded-up robust features (SURF) operator is an improved algorithm proposed on the basis of maintaining the excellent performance of the SIFT operator. SURF solves the shortcomings of high computational complexity and time-consuming SIFT [18]. The extraction and its feature vector description have been improved, and the calculation
speed has been greatly improved. SURF uses the Hessian matrix to stably obtain the extreme value of the image space. However, in the stage of finding the main direction, it completely depends on the gradient direction of the local pixel points, which may cause the main direction to be inaccurate [19], and because the feature vector extraction and matching are completely dependent on the above, the main direction is determined by the step and even a small deviation angle may cause the error of feature matching [20].

3. Two-Channel Convolutional Network Image Feature Extraction

The training idea of the convolutional neural network is to define a loss function, train the network through the existing samples, optimize the network parameters by using the back-propagation algorithm, and obtain the parameters of the network when the loss function is the smallest.

The neural network defines the loss function for a single sample \((x, y)\) as

\[
J(W, b; x, y) = \frac{1}{2} \| h_{w,b}(x) - y \|^2.
\]

Here, \(W\) and \(b\) are the weight and bias parameters of the network respectively, \(h_{w,b}(x)\) is the output of the sample \(x\) through the neural network, and \(y\) is the expected output value of the sample.

For a training set \(\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}\) with \(m\) samples, the overall loss function is

\[
J(W, b) = \frac{1}{m} \sum_{i=1}^{m} J(W, b; x^{(i)}, y^{(i)}) + \frac{\lambda}{2} \| W \|^2.
\]

(2)

According to the chain method, it is derived as

\[
\frac{\partial J(W, b; x, y)}{\partial W^{(l)}} = \frac{\partial J(W, b; x, y)}{\partial z^{(l+1)}} \cdot \frac{\partial z^{(l+1)}}{\partial W^{(l)}}.
\]

\[
\frac{\partial J(W, b; x, y)}{\partial W^{(l)}} = \frac{\partial J(W, b; x, y)}{\partial z^{(l+1)}} \cdot \frac{\partial w^{(l)} + b^{(l)}}{\partial W^{(l)}}.
\]

\[
\frac{\partial J(W, b; x, y)}{\partial W^{(l)}} = \frac{\partial J(W, b; x, y)}{\partial z^{(l+1)}} \cdot a^{(l)}.
\]

(7)

If the residual of the \(l\)-th layer network is \(\delta^{(l)} = \frac{\partial J(W, b; x, y)}{\partial z^{(l)}}\), then the residual of the output layer \((n)-th layer\) is

\[
\delta^{(n)} = \frac{\partial}{\partial z^{(n)}} \| h_{w,b}(x) - y \|^2 = \left( f(z^{(n)}) - y \right) f'(z^{(n)}).
\]

(8)

Using mathematical induction, the residual of the \(l\)-th layer is obtained, and the expression is

\[
\delta^{(l)} = \frac{\partial J(W, b; x, y)}{\partial z^{(l)}} = (w^{(l)})^T \cdot \delta^{(l+1)}.
\]

(9)

Therefore, the partial derivative of \(J(W, b)\) with respect to \(W\) is obtained as

\[
\frac{\partial J(W, b; x, y)}{\partial W^{(l)}} = \delta^{(l+1)} \cdot a^{(l)}.
\]

(10)

Simultaneously, the partial derivative of \(J(W, b)\) with respect to \(b\) can be obtained as

\[
\frac{\partial J(W, b; x, y)}{\partial b^{(l)}} = \delta^{(l+1)}.
\]

(11)

After the partial derivatives of \(J(W, b)\) with respect to \(W\) and \(b\) are obtained, the parameters of the weight \(W\) and the bias \(b\) can be updated and optimized according to (3) and (4).
Due to the introduction of convolutional layers and pooling layers, the training of CNN cannot be directly applied to the training method of DNN. There are several differences between them:

1. There is no activation function in the pooling layer, which can be solved by setting the activation function of this layer to \( f(z) = z \).
2. The pooling layer compresses the feature map during forward propagation.
3. DNN is a fully connected network, the output of the current layer is directly obtained by matrix multiplication, and the convolution layer is obtained by summing several matrices to obtain the current output.
4. For the convolutional layer, the operation method used by \( W \) is convolution.

In view of these differences, the backpropagation algorithm of CNN is discussed in the following two situations:

1. Knowing the \( \delta^{(l)} \) of the pooling layer, the \( \delta^{(l-1)} \) of the previous hidden layer is derived.

   The pooling layer first restores \( \delta^{(l)} \) all submatrices to the size before pooling during backpropagation. If maximum pooling is used, the value of the \( \delta^{(l)} \) submatrix corresponding to the pooling area is placed in the position where the maximum value is taken during the forward propagation. If average pooling is used, the values of the pooled area corresponding to the \( \delta^{(l)} \) submatrix are averaged and put into the restored matrix. This restoration process is defined as upsample. The restoration process is further explained below with an example.

   We assume that the pooling window size is 22, the window sliding step is 2, and the \( k \)-th submatrix of \( \delta^{(l)} \) is \( \delta_k^{(l)} = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix} \), and the submatrix is restored to the size before pooling and becomes \( \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 4 \\ 0 & 6 & 8 \\ 0 & 0 & 0 \end{pmatrix} \). If it is maximum pooling, we assume that the maximum positions are the lower left, lower right, upper right, and lower right during forward propagation, then the restored matrix is \( \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{pmatrix} \). If it is average pooling, the matrix after reduction is \( \begin{pmatrix} 0.5 & 0.5 & 1 & 1 \\ 0.5 & 0.5 & 1 & 1 \\ 1.5 & 1.5 & 2 & 2 \\ 1.5 & 1.5 & 2 & 2 \end{pmatrix} \), the value of the upper layer \( \frac{\partial J(W, b; x, y)}{\partial a_k^{(l-1)}} \) can be obtained by upsample, then there are

\[
\delta_k^{(l-1)} = \text{upsample}(\delta_k^{(l)}) \cdot f'(z^{(l-1)}).
\]

   For the tensor \( \delta^{(l-1)} \), we have

\[
\delta^{(l-1)} = \text{upsample}(\delta^{(l)}) \cdot f'(z^{(l-1)}).
\]

2. Knowing the \( \delta^{(l)} \) of the convolutional layer, it is deduced to \( \delta^{(l-1)} \) of the previous hidden layer.

   In CNN, the recurrence formula of \( \delta^{(l)} \) and \( \delta^{(l-1)} \) is

\[
\delta^{(l)} = \frac{\partial J(W, b; x, y)}{\partial z^{(l-1)}} = \frac{\partial J(W, b; x, y)}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial z^{(l-1)}} = \delta^{(l)} \frac{\partial z^{(l)}}{\partial z^{(l-1)}}.
\]

   According to the forward propagation formula of the convolutional layer, we get

\[
z^{(l)} = a^{(l)} \ast W^{(l)} + b^{(l)} = f(z^{(l-1)}) \ast W^{(l)} + b^{(l)}.
\]

   Then, there are

\[
\delta^{(l-1)} = \delta^{(l)} \ast \frac{\partial z^{(l)}}{\partial z^{(l-1)}} = \delta^{(l)} \ast \text{rot180}(W^{(l)}) \cdot f'(z^{(l-1)}).
\]

   Among them, \( \text{rot180}(W) \) means to rotate 180 degrees.

   Now, given the \( \delta^{(l)} \) of the convolutional layer, the gradient of this layer \( W \) is...
\[
\frac{\partial J(W, b; x, y)}{\partial W^{(l)}} = \frac{\partial J(W, b; x, y)}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial W^{(l)}} = a^{(l-1)} \ast \delta^{(l)}. \tag{17}
\]

Since \( \delta^{(l)} \) is a three-dimensional tensor and \( b \) is a one-dimensional vector, the algorithm sums the submatrix items of \( \delta^{(l)} \) respectively, and the obtained one-dimensional error vector is the gradient of \( b \). The calculation formula is
\[
\frac{\partial J(W, b; x, y)}{\partial b^{(l)}} \sum_{u,y}(\delta^{(l)})u, y. \tag{18}
\]

The original LBP operator is defined in a fixed rectangular neighborhood of \( 3 \times 3 \), which cannot meet the needs of textures of different sizes, so a circular LBP operator is generated. Compared with the original LBP operator, the circular LBP operator has two improvements: (1) the fixed neighborhood of 33 is extended to an arbitrary neighborhood; (2) the square neighborhood is no longer used, and the circular neighborhood is adopted. The circular LBP operator can be expressed as \( \text{LBPP}, R \), where \( R \) is the radius of the circular neighborhood and \( P \) is the number of sampling points. Several common LBP operators are shown in Figure 1.

The calculation formula of the circular LBP operator is
\[
\text{LBP}_{P,R} = \sum_{i=1}^{P} s(g_i - g_c)2^{i-1}. \tag{19}
\]

Here, \( s(u) = \begin{cases} 1, u \geq 0 \\ 0, \text{other} \end{cases} \), \( g_c \) is the gray value of the pixel in the center of the neighborhood, and \( g_i \) is the gray value of the pixel of the \( i \)-th sampling point in the neighborhood. To calculate the pixel gray value of the \( i \)-th sampling point, the coordinates \( (x_i, y_i) \) of the \( i \)-th sampling point are first calculated, and the calculation formula is
\[
\begin{align*}
x_i &= x_c + R \cos \left( \frac{2\pi i}{P} \right), \\
y_i &= y_c + R \sin \left( \frac{2\pi i}{P} \right).
\end{align*} \tag{20}
\]

Here, \((x_c, y_c)\) is the coordinate of the center pixel. It can be seen from Figure 1 that some sampling points on the boundary of the circular neighborhood cannot just fall within the pixel grid, and may also fall on the boundary. At this time, the pixel value of the sampling point is calculated by the bilinear interpolation method, and the calculation formula is
\[
g(x_i, y_i) = [1-x_i] \begin{bmatrix} g(x_{i-1}, y_i) & g(x_{i-1}, y_{i-1}) \end{bmatrix} \begin{bmatrix} 1-y_i \\ y_i \end{bmatrix}. \tag{21}
\]

In order to reduce the variety of binary patterns, Ojala proposes an equivalent pattern, which is also called uniform LBP. The main idea is that if the binary string corresponding to the mode does not jump from 0 to 1 or from 1 to 0 more
than twice, then the mode is an equivalent mode, and all other modes are mixed modes, which are expressed as

$$U(LBP_{P,R}) = \sum_{i=1}^{P} |s(g_i - g_e) - s(g_{i-1} - g_e)|.$$  \hfill (22)

Here, $g_0 = g_P$, and the mode satisfying $U \leq 2$ is the equivalent mode. The improved mode types are reduced from the original $2^P$ to $P^2 (P - 1) + 2$, which not only does not lose image information, but also reduces the impact of high-frequency noise.

Three-dimensional convolution (3D convolution) performs convolution operations on both the spatial and temporal domains by extending the convolution kernel to the temporal domain. In this way, it can extract both spatial information of a single frame of image and temporal information between adjacent video frames. 3D convolution combines multiple consecutive video frame images into a cube and slides the 3D convolution kernel on the cube to perform the convolution operation. When performing the convolution operations on a sequence of video frames, the difference between 2D convolution and 3D convolution is as follows: after a 2D convolution, the generated feature map is an image, which loses the time information of the input data. However, through 3D convolution, the generated feature map is still a sequence of feature maps, which effectively captures the motion information of the target. The convolution process of 2D convolution and 3D convolution on the video frame sequence is shown in Figure 2.

The network used in this section improves the original C3D network model and retains two fully connected layers, and its network structure is shown in Figure 3. The improved C3D network contains 5 convolutional layers, 5 pooling layers, 2 fully connected layers, and 1 Softmax classifier. A ReLU layer is added after each convolutional layer and the first fully connected layer in the network. This is because there is a simple linear relationship between the layers of the network. By introducing a nonlinear function as an activation function, the algorithm increases the nonlinear relationship between the layers so that the network can fit complex functions. The formula of the ReLU function is shown in equation (5).

$$f(x) = \max(0, x).$$  \hfill (23)
ResNet adopts a residual structure, as shown in Figure 4(a). This module has two branches, one of which is the normal convolutional layer output and the other that directly connects the input to the output. The final output of the module is the arithmetic sum of the two branches, which is formulated as

$$H(x) = F(x) + x. \quad (24)$$

Here, $H(x)$ represents the output of the entire structure, $x$ is the input, and $F(x)$ is the output of the convolutional layer. ResNet defines a residual function $F(x) = H(x) - x$. When all parameters in the $F(x)$ branch are 0, $H(x) = x$ is the identity map. ResNet no longer learns the output of the entire structure, it is changed to learn the difference between the target value $H(x)$ and the input $x$, and the training goal is to make the residual function $F(x)$ approach 0.

Obviously, fitting the residual function is easier than fitting the identity mapping function, and the training process of the network is optimized. The residual structure is implemented by connecting the forward neural network and the identity map without introducing additional parameters. This will not increase the computational complexity of the network, and the network training method still uses backpropagation. After adopting the residual structure, the deep neural network achieves the ideal classification effect. A typical residual module structure is shown in Figure 4(b), which consists of stacking convolutional layers. Among them, $3 \times 3 \times 3$ is the size of the convolution kernel of this layer, 128 is the number of convolution kernels, and /1 × 2 × 2 indicates that the
convolution kernel moves with a step size of 1 in the time domain and 2 in the spatial domain.

It is usually implemented by $1 \times 1$ convolution, and the formula is

$$H(x) = F(x) + Wx.$$  \hfill (25)

It should be noted that scheme (1) is not applicable to the case where the output feature map of the convolutional layer shown in Figure 4 is halved with respect to the size of the identity map feature map and the number of channels is doubled. Therefore, the dashed connection of the identity mapping in the R3D network structure adopted in this paper adopts the method in scheme (2). By adding a convolutional layer with a convolution kernel size of $1 \times 1 \times 1$ and a convolution stride of $1 \times 2 \times 2$ or $2 \times 2 \times 2$, the input feature map size and number of channels of the two branches are kept consistent.

The structure of the R3D model used in this paper is shown in Figure 5.

The 3D convolution operation simultaneously extracts video spatial and temporal dimensions through a 3D convolution kernel. According to the three-dimensional convolution properties, spatial modeling and temporal modeling can be decomposed into two separate steps, which are replaced by two-dimensional spatial convolution (2D convolution) and one-dimensional temporal convolution (1D convolution).

---

**Figure 9**: System design diagram. (a) Calligraphy identification block diagram. (b) System structure diagram.
This process is called three-dimensional convolution decomposition, which is also called (2+1)D decomposition vividly.

The 3D convolutional network preserves both temporal and spatial information through layer-by-layer transfer. In this paper, a video clip is fed into a 3D convolutional neural network. We assume that the tensor generated by the $i$-th convolutional layer is $z_i$, then $z_i$ is a four-dimensional tensor of size $N_i \times L \times H_i \times W_i$, where $N_i$ is the number of convolution kernels of the $i$-th convolutional layer, $L$ is the time dimension of the feature map, and $H_i \times W_i$ is the spatial dimension of the feature map. Each convolution kernel is a four-dimensional tensor of size $N_{i-1} \times t \times d \times d$, where $N_{i-1}$ is the number of convolution kernels of the $i-1$-th convolutional layer, $t$ is the size of the three-dimensional convolution time dimension, and $d \times d$ is the size of the three-dimensional convolution space dimension.

The $(2+1)$D decomposition is a decomposition of a three-dimensional convolution kernel with $N_i$ tensor size $N_{i-1} \times t \times d \times d$ into a two-dimensional spatial convolution kernel with $M_i$ tensor size $N_{i-1} \times 1 \times d \times d$ and a one-dimensional temporal convolution kernel with $N_i$ tensor size $M_i \times t \times 1 \times 1$. The hyperparameter $M_i$ determines the subspace dimension of the feature map between spatial convolution and temporal convolution. In order to keep the parameters of the network before and after decomposition unchanged, the calculation formula of $M_i$ is as follows:

$$M_i = \frac{td^2N_{i-1}N_i}{d^2N_{i-1} + tN_i}$$  \hspace{1cm} (26)$$

Figure 6 shows the 3D decomposition process when the input tensor $z_{i-1}$ is a single channel (that is, $z_{i-1}$). If the 3D convolution has a span in space or time (implementing downsampling), it should be decomposed in the spatial and temporal dimensions accordingly.

Compared with the 3D convolution, the $(2+1)$D decomposed convolution module does not reduce the number of parameters, but a ReLU layer is added between the 2D convolution and the 1D convolution. This leads to an increase in the number of nonlinear functions in the network, allowing the network to fit more complex functions. The above-mentioned $(2+1)$D decomposition process does not change the number of parameters of the network, and the server memory requirements are relatively high when training the network.

This paper improves on the above $(2+1)$D decomposition. Specifically, this paper directly decomposes $N_i$ three-dimensional convolution kernels of size $t \times d \times d$ into $N_i$ two-dimensional spatial convolution kernels of size $1 \times d \times d$ and $N_i$ one-dimensional temporal convolution kernels of size $t \times 1 \times 1$. The improved $(2+1)$D decomposition can greatly reduce the network parameters and speed up the network operation.

The R $(2+1)$D network used in this paper is a network formed by decomposing the convolutional layer according
to the improved (2 + 1)D decomposition process based on the R3D network used in the previous section. The three-dimensional convolution kernel of $3 \times 3 \times 3$ is decomposed into a two-dimensional spatial convolution kernel of $1 \times 3 \times 3$ and a one-dimensional temporal convolution kernel of $3 \times 1 \times 1$. The specific decomposition process is shown in Figure 7, where Conv_s represents the decomposition in the spatial domain and Conv_t represents the decomposition in the time domain.

The structure diagram of the R (2 + 1) D network model is shown in Figure 8.

4. A Two-Channel Convolutional Network for Calligraphic Style Feature Extraction and Discrimination

The third part constructs a calligraphy style feature extraction and identification system based on a two-channel convolutional network. The authenticity identification process of calligraphy works is generally divided into four steps: scanning of original works, preprocessing of original works, extraction of feature points, comparison and matching of authentic and fake works, and performance evaluation of the identification system. The general process of calligraphy identification is shown in Figure 7, where Conv_s represents the decomposition in the spatial domain and Conv_t represents the decomposition in the time domain.

The structure diagram of the R (2 + 1) D network model is shown in Figure 8.

Table 1: Performance verification of the calligraphy style feature extraction and identification system based on the two-channel convolutional neural network.

| Number | Feature extraction | Calligraphy identification |
|--------|-------------------|---------------------------|
| 1      | 95.40             | 88.44                     |
| 2      | 93.31             | 91.18                     |
| 3      | 92.42             | 91.00                     |
| 4      | 93.74             | 90.42                     |
| 5      | 91.26             | 91.14                     |
| 6      | 91.88             | 92.32                     |
| 7      | 95.84             | 91.39                     |
| 8      | 94.49             | 87.94                     |
| 9      | 90.50             | 88.17                     |
| 10     | 95.33             | 92.67                     |
| 11     | 90.11             | 87.35                     |
| 12     | 93.07             | 87.78                     |
| 13     | 95.03             | 91.10                     |
| 14     | 93.77             | 92.76                     |
| 15     | 94.71             | 88.93                     |
| 16     | 91.53             | 90.41                     |
| 17     | 90.82             | 89.42                     |
| 18     | 92.09             | 90.69                     |
| 19     | 94.11             | 91.13                     |
| 20     | 94.58             | 91.99                     |
| 21     | 92.45             | 86.25                     |
| 22     | 95.25             | 88.49                     |

5. Conclusion

The contour of calligraphy is also called edge detection, and the result of extraction is several closed contour curves formed by contour tracking. Contour feature extraction is a common method in image processing. When extracting the features of calligraphy characters, the algorithm calculates the feature weight and the authenticity judgment function and constructs the authenticity identification system. Figure 9(b) is the structure diagram of the authenticity identification system.

Figure 10 shows an example of calligraphy image recognition proposed in this paper.

On the basis of the above research, the effect of the calligraphy style feature extraction and identification system based on the dual-channel convolutional neural network proposed in this study is verified, and the results are shown in Table 1.

From the above research, it can be seen that the calligraphy style feature extraction and identification system based on the dual-channel convolutional neural network proposed in this paper has a good performance in calligraphy style feature extraction and identification.
proposes a calligraphy style feature extraction and identification technology based on a two-channel convolutional neural network, constructs an intelligent calligraphy-style feature extraction and identification system, and improves the effect of calligraphy style feature extraction and identification. The research shows that the calligraphy style feature extraction and identification system based on the two-channel convolutional neural network proposed in this paper has a good performance in calligraphy style feature extraction and identification.

Data Availability

The labeled dataset used to support the findings of this study is available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This study was sponsored by Xi’an Jiaotong University.

References

[1] M. H. Aghdam and S. Heidari, “Feature selection using particle swarm optimization IN text categorization[1],” Journal of Artificial Intelligence and Soft Computing Research, vol. 5, no. 4, pp. 38–43, 2015.

[2] LiDe, Z Xue, and LiH. Cui, “Text recognition algorithm based on text features[1],” International Journal of Multimedia & Ubiquitous Engineering, vol. 11, no. 5, pp. 209–220, 2016.

[3] S. T Gissel, “Scaffolding students’ independent decoding of unfamiliar text with a prototype of an eBook-feature,” Journal of Information Technology Education: Research, vol. 14, no. 2015, pp. 439–470, 2015.

[4] R. R. Kamble and D. V. Kodavade, “Relevance feature search for text mining using FCustering algorithm,” International Journal of Computer Science and Engineering, vol. 6, no. 7, pp. 223–227, 2018.

[5] M. Pandi and V. D. K. Rajendran, “Efficient feature extraction for text mining[1],” Advances in Natural & Applied Sciences, vol. 10, no. 4, pp. 64–73, 2016.

[6] M. Mojaveriyan, H. Ebrahimpourkomleh, H. Ebrahimpourkomleh, S. J. Mousavirad, and M. Seyed jalaleddin, “IGICA: a hybrid feature selection approach in text categorization,” International Journal of Intelligent Systems and Applications, vol. 8, no. 3, pp. 42–47, 2016.

[7] M. Nayak and A. K. Nayak, “Odia running text recognition using moment-based feature extraction and mean distance classification technique,” Advances in Intelligent Systems and Computing, vol. 309, no. 3, pp. 497–506, 2015.

[8] E. Ramalakshmi and S. Golla, “An advanced fuzzy constructing algorithm for feature discovery in text mining,” International Journal of Computer Application, vol. 127, no. 17, pp. 30–34, 2015.

[9] M. Soleymanpour and H. Marvi, “Text-independent speaker identification based on selection of the most similar feature vectors[1],” International Journal of Speech Technology, vol. 20, no. 1, pp. 1–10, 2016.

[10] T. Oki, “Scene text localization using object detection based on filtered feature channels and crosswise region merging,” Jf, growth & change, vol. 21, no. 3, pp. 61–76, 2015.

[11] Z Tehseen, M. P. Akhter, and Q. Abbas, “Comparative study of feature selection approaches for Urdu text categorization,” [J], malaysian journal of computer science, vol. 28, no. 2, pp. 93–109, 2015.

[12] T. Zia, Q. Abbas, Q. Abbas, and M. P. Akhter, “Evaluation of feature selection approaches for Urdu text categorization,” International Journal of Intelligent Systems and Applications, vol. 7, no. 6, pp. 33–40, 2015.

[13] A. Tommasel and D. Godoy, “A social-aware online short-text feature selection technique for social media,” Information Fusion, vol. 40, no. 1, pp. 1–17, 2017.

[14] G. Wu, M. Zhao, and L. Han, “A fingerprint feature extraction algorithm based on optimal decision for text copy detection,” International Journal of Security and Its Applications, vol. 10, no. 11, pp. 67–78, 2016.

[15] K. Yan, Z. Li, and C. Zhang, “A New multi-instance multi-label learning approach for image and text classification,” Multimedia Tools and Applications, vol. 75, no. 13, pp. 7875–7890, 2016.

[16] Z. Robati, M. Zahedi, N. Fayazi Far, and F. Najmeh Fayazi, “Feature selection and reduction for Persian text classification,” International Journal of Computer Application, vol. 109, no. 17, pp. 1–5, 2015.

[17] G. Kumar and K. Vivekanandan, “Intelligent model view controller based semantic w call through mishmash text featuring technique,” Journal of Computational and Theoretical Nanoscience, vol. 14, no. 4, pp. 2021–2029, 2017.

[18] B. Shi, X. Bai, and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298–2304, 2017.

[19] Al-Tahraw, "Polynomial neural networks versus other Arabic text classifiers," Journal of Software, vol. 11, no. 5, pp. 418–430, 2016.

[20] Y. Hu, E. E. Milios, and J. Blustein, “Document clustering with dual supervision through feature reweighting,” Computational Intelligence, vol. 32, no. 3, pp. 480–513, 2016.