Hindawi
Complexity
Volume 2021, Article ID 6617799, 11 pages
https://doi.org/10.1155/2021/6617799

Research Article

An Improved Deep Learning Network Structure for Multitask Text Implication Translation Character Recognition

Xiaoli Ma, Hongyan Xu, Xiaoqian Zhang, and Haoyong Wang

Department of Foreign Language, Hebei Agricultural University, Baoding, Hebei 071001, China

Correspondence should be addressed to Haoyong Wang; wgywhy@hebau.edu.cn

Received 11 December 2020; Revised 30 January 2021; Accepted 4 February 2021; Published 12 February 2021

Academic Editor: Wei Wang

With the rapid development of artificial intelligence technology, multitasking textual translation has attracted more and more attention. Especially after the application of deep learning technology, the performance of multitask translation text detection and recognition has been greatly improved. However, because multitasking contains the interference problem faced by the translated text, there is a big gap between recognition performance and actual application requirements. Aiming at multitasking and translation text detection, this paper proposes a text localization method based on multichannel multiscale detection of the largest stable extreme value region and cascade filtering. This paper selects the appropriate color channel and scale to extract the maximum stable extreme value area as the character candidate area and designs a cascaded filter from coarse to fine to remove false detections. The coarse filter is based on some simple morphological features and stroke width features, and the fine filter is trained by a two-recognition convolutional neural network. The remaining character candidate regions are merged into horizontal or multidirectional character strings through the graph model. The experimental results on the text data set prove the effectiveness of the improved deep learning network character model and the feasibility of the textual implication translation analysis method based on this model. Among them, the text contains translation character recognition results prove that the model has good description ability. The characteristics of the model determine that this method is not sensitive to the scale of the sliding window, so it performs better than the existing typical methods in retrieval tasks.

1. Introduction

It is a very challenging topic to detect the text contained in multitask translation, instead of traditional scanning paper, business cards, ID cards, etc., and it also has high application value [1, 2]. It is a lot of computer applications based on vision. Text belongs to relatively high-level semantic information in visual information, and it plays a great role in the understanding of the translation content contained in the text [3]. In addition to traditional pixel, color, and structural features, text information also has clear and targeted semantic information. In the field of computer vision, in addition to some low-level features such as texture, edge lines, corner points, etc., it is more important to describe text information by combining high-level semantics with low-level features. There is a large amount of text information in multitask implication translation, and these kinds of text information play a very good supplementary role in the expression of the translation content of the text implication [4]. If the text information content can be obtained from the multitask implied translation text, the text can be understood in higher semantics.

Text recognition extracts text features after preprocessing the text and recognizes the text information in the text based on these features, so as to provide some necessary semantic information for text analysis and environmental perception [5]. Compared with text positioning and segmentation, there is less attention to recognition, mainly because the current optical character recognition (OCR) recognition technology is very mature. Many researchers send the results of positioning and segmentation directly into OCR software for recognition after some preprocessing [6]. The expected goal can basically be achieved. At present, it is mainly divided into two categories, one is to recognize through OCR software, and the other is to use some search strategies to directly detect candidate characters in the text and then combine with graph matching or some discriminant models. For example, the CRF model completes the recognition of candidate characters [7]. Relevant scholars extract SIFT features from the collected text, then directly compare them with the template characters in the dictionary set, and then use voting criteria and geometric verification methods to modify the comparison results to obtain the final recognition result [8]. Researchers use Gaussian filters or basic scales and rough scales to improve HOG features, extract improved HOG features from the collected text, and combine some recognizers to recognize characters in the text [9, 10]. The accuracy of text segmentation will directly affect the accuracy of text recognition. Text segmentation technology is also an important part of multitask translation text recognition. Text segmentation refers to separating characters one by one from the positioned text line. The text segmentation technology of multitask containing translated text mainly involves two contents: separating text and background, and segmenting text lines. Multitasking implies that the key content of translation text segmentation is the separation of text and background [11]. There are two main types of text line segmentation techniques. The first category is based on connected domain segmentation or projection method segmentation. The technical core of this type of method is that a single character is basically a connected area, and there is often a character spacing between different characters, and the text is divided into single characters [12]. The second category is the use of recognized character segmentation methods. This type of segmentation method divides the text line into individual characters through the language characteristics of the characters [13]. The sliding window mechanism is adopted to divide the text line into multiple character combinations, and the recognition result is used to find the most reasonable dividing line. However, the same prerequisite for these two types of segmentation methods is the need to separate the text line from the background [14]. The recognizers used for character recognition include structural feature-based recognizers, support vector machines, convolutional neural networks, random forests, and AdaBoost recognizers [15, 16]. Similar to the extraction of structural features, the recognizer using structure is formed on the basis of structural features [17]. This type of recognizer is characterized by high accuracy and poor robustness. The recognition model of support vector machine is still used very frequently in recent research. The main reason is that SVM has good effects on pattern recognition, regression problems, and feature selection. Compared with other recognizers, it has better robustness. The convolutional neural network recognition model has super high accuracy in recognition problems, which can be said to open a new journey of artificial intelligence [18]. The random forest model is not very common in recognition and recognition, but as a representative of strong recognizer, it has strong recognition ability [19]. Its core lies in multiple decision trees to make judgments without overfitting. The AdaBoost recognizer is also one of the commonly used strong recognizers [20]. Combining the description of multiple features, each weak recognizer is voted by voting to obtain a higher accuracy rate.

Different from traditional text documents, multitasking implies that the translated text has the characteristics of different font shapes and colors, complex and changeable backgrounds, and numerous interferences. In this paper, combined with deep learning technology, the traditional multitask implication translation text detection method is improved, and a multitask implication translation text detection method based on multichannel, multiscale, and cascade filtering is proposed. We extract the maximum stable extreme value area as the character candidate area under the appropriate channel and scale. A cascaded filter from coarse to fine is designed to remove false detections. The coarse filter is based on some simple morphological features, and the fine filter is performed by a well-trained two-recognition convolutional neural network. The remaining character candidate regions are merged into character strings through a graph model. Experimental results show that this method can effectively model characters. The model is a production model, and the inherent advantages of the production model make it perform well in retrieval tasks. In addition, because the model is based on local features and independently describes the structural characteristics of characters, the model has more obvious performance advantages in structural characters.

The rest of this article is organized as follows. Section 2 discusses related theories and technologies. Section 3 builds a deep learning network model based on multichannel, multiscale, and cascade filtering. Section 4 analyzes the experimental results. Section 5 summarizes the full text.

2. Related Theories and Technologies

2.1. Machine Learning. Among the big concepts of artificial intelligence, machine learning is the most commonly used and widely used. There are three main research directions in artificial intelligence: the semiotic school that tries to simulate the human mind, the connection school that simulates the brain structure, and the behavior school that simulates human behavior. This corresponds to three different research results: knowledge representation, neural network, and intelligent robot.

Machine learning is divided into three methods: supervised machine learning, unsupervised machine learning, and semisupervised machine learning. They are different in principle and structure, and their characteristics are also different. Supervised learning is a common technique widely used in neural network training. Supervised learning needs to learn a function (model parameter) from a given training data set. The purpose is that when there are new input data, the supervised learning network can predict the output data result based on the trained model parameters. Compared with supervised learning, the input data of unsupervised learning are not labeled, and the sample data category is unknown, which means that it is not possible to train a model parameter that can be used to predict the result like supervised learning, but it needs to be based on the input data. The similarity identifies the sample set. Unlike supervised learning, which has its own training set and test samples, unsupervised learning needs to find the distinction between classes in the only input data.

Compared with the above two learning methods, the main application scope of semisupervised learning is slightly different. It is mainly applied to the situation where the input data have a small part of the identification mark, and most of the data are not marked, but they do not meet the requirements of manual marking. Therefore, when using semisupervised learning, the core idea is to design a specific query algorithm to query part of the data according to some specific conditions and ask experts to mark it and finally use the queried sample data for identification.

In supervised learning, the more training samples, the more accurate feature values can be extracted. In other words, the more training samples, the better the final result. At this stage, most applications in the AI field are still based on supervised learning. Unsupervised and semisupervised learning need to be further developed.

2.2. Artificial Neural Network. Artificial neural network is an important method for realizing machine learning. Although it has been proposed for many years, researchers around the world are still constantly innovating and constructing new structures and algorithms to achieve the goal of better realization of artificial intelligence.

Artificial neural network is a network composed of a large number of structural connections similar to neurons. During the construction process, the field of artificial intelligence is organically combined with modern biology and modern neuroscience, and it simplifies the basic characteristics of the human brain. Biological neurons are the most basic unit of brain tissue, including the body, axons, dendrites, synapses, and other parts. When there is a signal input from the outside world, the excitement is transferred from the synapse to the dendrites, analyzed by the body cell, and transmitted along the axon to the next neuron cell. Similar to biological neurons, artificial neurons are also the most basic component of artificial neural networks, as shown in Figure 1. When there is an excitement or inhibition signal input from the outside, it will be analyzed by a weighted function that has the effect of the cell body. If the output exceeds the threshold, the excitement or inhibition signal will be transmitted to the next layer of artificial neurons.

The artificial neural network has three elements, namely, the neuron characteristics of the network model, the topology structure, and the training rules. Once these three elements are determined, a specific artificial neural network model is also determined. The neuron model mentioned above is the basic unit of the artificial neural network, and the neuron characteristics are unique to the neuron model. It contains three basic elements, namely, a set of connection weights, a summation unit, and a nonlinear activation function.

The neural network structure is composed of a large number of neurons, which is the main difference between different neural networks. From the connection method, it is mainly divided into two types: feedforward neural network and feedback neural network. The feedforward neural network is a widely used neural network. Its neurons are arranged hierarchically, the neurons between layers are connected to each other, and the neurons in the same layer are parallel and unconnected and are divided into input layers according to the functions of the input layer and hidden layer. The main function of the input layer is to input external data and transmit the input data to one or more hidden layers for processing; after the hidden layer processing, data with more distinctive features are transmitted to the output layer as output data for the next step data processing. Generally speaking, the single hidden layer structure is the most common; the three-layer or four-layer network occasionally appears, mainly for special data processing, and the more layer structure is generally not done due to problems such as processing time and low efficiency use. The feedback network is different from the feedforward network. In its structure, any two neurons may be connected, and each neuron can perform the task of input and output. The input data are passed through each neuron in the network.

The training rules of neural networks can generally be called learning methods, which are mainly divided into the three learning methods mentioned above. Each has its advantages and disadvantages, but the purpose is to achieve the purpose of processing data by adjusting its own parameters according to a certain predetermined measurement.

Compared with traditional computers, artificial neural networks have certain advantages in processing information and input data.

2.2.1. Parallelism. Compared with the serial method used by traditional computers, the neurons of the artificial neural network are connected in parallel, which greatly improves the computing power and efficiency.

2.2.2. Self-Organization. Different from the traditional computer processing system, the neural network can continuously adjust the parameters of its own network in the process of processing data and finally achieve the goal of maximizing efficiency.

2.2.3. Robustness and Fault Tolerance. In a neural network, all parameters are distributed among each neuron. Once a neuron has a fault problem, it will only reduce the performance and processing efficiency of a part of the network and will not paralyze the entire network.

2.3. Convolutional Neural Network. Convolutional neural network is currently the most widely used deep learning network. It was first used in handwriting recognition and achieved very good results. Compared with other ordinary neural networks before, the convolutional neural network contains a feature extractor composed of a convolutional layer and a subsampling layer. In each convolutional layer, all neurons in each feature plane share weights, that is, the convolution kernel of the convolutional layer. In the process of training the network, the weight value changes from small

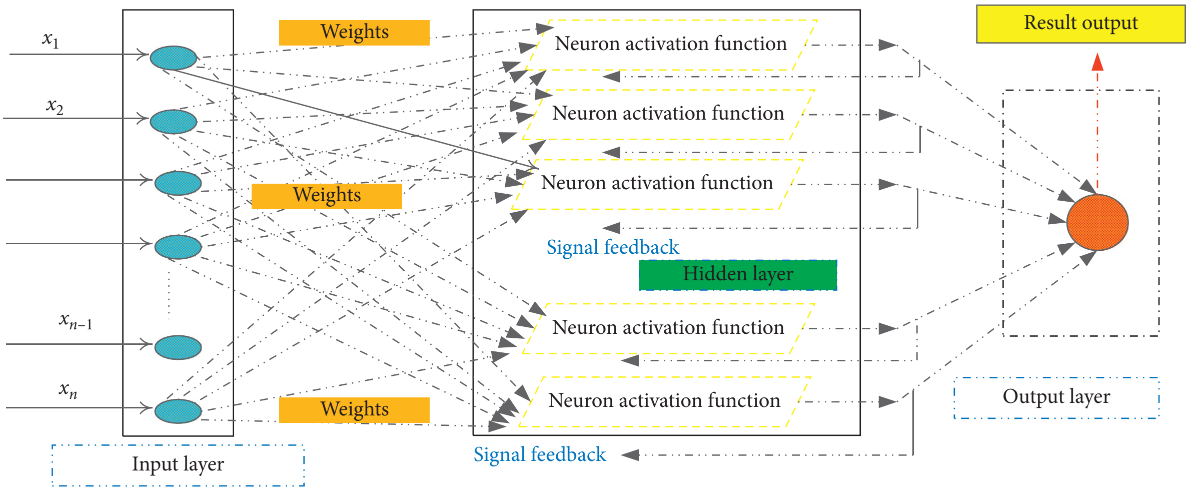

x_1
x_2
x_{n-1}
x_n
Weights
Weights
Weights
Input layer
Neuron activation function
Neuron activation function
Neuron activation function
Signal feedback
Hidden layer
Neuron activation function
Neuron activation function
Signal feedback
Result output
Output layer

Figure 1: Neuron model structure.

to large, and finally, a value suitable for data feature extraction will be obtained, and neurons in the same layer share this value, thus reducing the connection between different layers.

Convolutional neural networks include input layer, convolutional layer, pooling layer, fully connected layer, and output layer. Among them, convolutional layer, pooling layer, and fully connected layer are generally collectively called the hidden layer, as shown in Figure 2. The main function of the convolutional layer is to extract the features in the input data, and through self-learning to control the parameters in the convolution kernel to extract the feature values, the weight sharing method greatly reduces the number of parameters and ensures the sparsity of the network. The function of the pooling layer is to reduce the dimensionality of the output data output by the convolution kernel, so that its size can be reduced while retaining the complete data features; the fully connected layer mainly plays a role of recognition and combines the features in the input data completely to the output layer.

2.3.1. Convolutional Layer. The convolutional layer is composed of multiple convolutional units. It is the core of the convolutional neural network, which is responsible for extracting the characteristics of the input data. It is also the realization part of the main idea of the convolutional neural network [21–23]. The convolutional layer obtains the feature value of the input data by convolution operation on the input information and outputs it after subsequent operations. Among them, the convolution kernel plays an important role as the core component of the convolution layer.

If the activation function part is ignored, the processed output data of each layer are a linear mapping of the input data. No matter how many convolutional layers are superimposed, the final output data are a linear mapping of the original input data. The result of the hidden layer in the middle is consistent, and the function of extracting feature values is lost.

2.3.2. Pooling Layer. The pooling layer is also called the downsampling layer and is usually used after the convolutional layer. Through the role of the convolutional layer, the size of the input data will greatly increase and it is difficult to perform the next operation [24, 25]. The purpose of the pooling layer is to reduce the size of the large-size data output from the convolutional layer, thereby reducing the parameters of the fully connected layer.

There are two main ways of pooling: maximum pooling and average pooling. The maximum pooling method is to take the maximum size value of the input data as the size value of all output data, and the average pooling method is to take the average value. After the effect of the pooling layer, the number of parameters in the output data will be significantly reduced compared to the input data, and the feature values will remain unchanged, preventing overfitting.

2.3.3. Fully Connected Layer. The fully connected layer generally appears in the last few layers of the structure. The neurons in this layer are connected to the neurons in the upper layer. The purpose is to integrate the local feature information obtained in the convolutional layer and the pooling layer.

2.4. Deep Residual Network. The residual network is similar to the convolutional neural network in the network structure, with the basic input layer, hidden layer, and output layer, as shown in Figure 3. However, unlike the convolutional neural network, the hidden layer of the residual network has a basic structure unique to the residual network, that is, the residual block. It consists of two convolutional layers and a pooling layer and other structures, which are connected using shortcuts. The input data can be processed in the convolution branch, and the other pooling branch directly transmits the input data to the output data. The real output data are obtained by adding up the outputs of the persons, which are output through the fully connected layer.

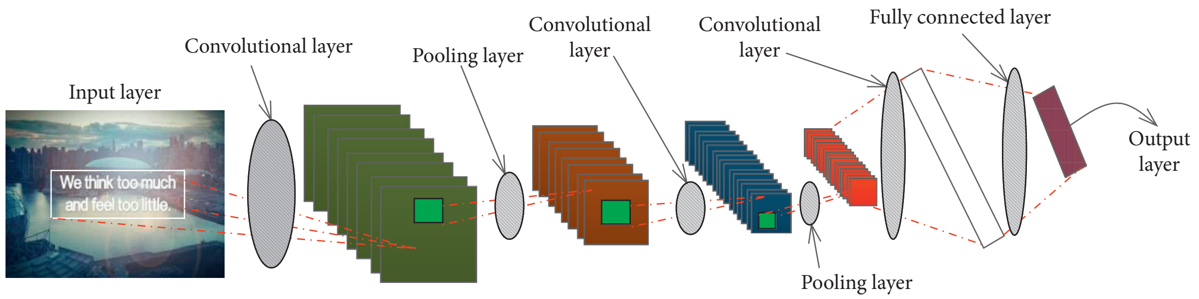


Figure 2: Working principle of convolutional neural network.

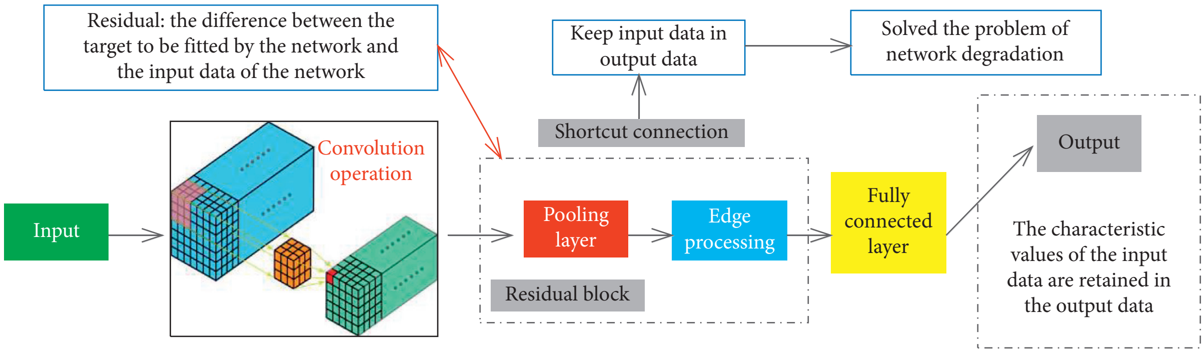

Residual: the difference between the target to be fitted by the network and the input data of the network
Keep input data in output data
Solved the problem of network degradation
Shortcut connection
Convolution operation
Input
Pooling layer
Edge processing
Residual block
Fully connected layer
Output
The characteristic values of the input data are retained in the output data

Figure 3: Block diagram of residual network structure.

Compared with the traditional network, the training goal of the residual network has changed. The original input data x are added to the original $f(x)$. This is also the most basic idea of the ResNet network. Reflected on the data, the characteristic value of the input data is still retained in the output data after a residual structure. Traditional convolutional networks will have the problems of gradient disappearance and network degradation when information is transmitted, resulting in the inability to continue training when the network structure is too deep. In order to solve this problem, ResNet directly takes the input information as part of the output information and retains the integrity of the characteristic information of the input information.

3. Deep Learning Network Model Based on Multichannel, Multiscale, and Cascade Filtering

3.1. Extraction of Character Candidate Regions Based on Multichannel and Multiscale. Multitasking implies that the color or gray-scale intensity between characters in a translation picture is often different, but for a single character, its gray-scale intensity or color is generally uniform, and the intensity difference between the background and the character is very large, so each character can be considered as a MSER, so MSER can be extracted as a character candidate area. However, MESR is based on pixel-level operations, so it is very sensitive to noise or damage to a single pixel. For a text detection system based on the connected component analysis method, the primary goal is to detect as many real character regions as possible, because it is difficult to recover previously lost characters in the subsequent processing. Here, we set the threshold Δ of MSER to a minimum value of 1, so that it can cope with various challenging situations as much as possible. Although a low threshold will introduce a large number of false detections, it makes the detector robust enough to detect various challenging characters, thereby ensuring a high character level recall rate and further ensuring the word level.

Traditional MSER detection is mostly for gray-scale text. The gray-scale channel is the weighted sum of color tones. When extracting character candidate regions under this channel, due to the various interferences in translation contained in multitasking, it is more robust difference. The contrast of some characters under the gray channel is relatively low, which is prone to false detection, missed detection, and incompleteness.

In the multitask implicit translation text, there are inevitably some special characters, such as large fonts, bitmap fonts, and low-contrast characters. These characters are difficult to use the MSER detector to extract complete characters at the original scale. In addition, due to low Δ value, a larger character is easily divided into several parts. For this reason, we propose to detect MSER as a character candidate area at a scale of 0.125 times. This method can effectively converge on the segmented areas and deal with lower contrast and translucent characters. In proportion, the detection speed at small scales is very fast, and the computational complexity will not increase too much. At a small scale, due to the scaling, the color of the characters becomes more uniform and the originally separated parts will merge together, which greatly reduces the influence of factors such as nonuniform illumination.

3.2. Coarse Filtering Based on Morphology. After multichannel, multiscale, and low Δ MSER detection, a large number of character candidate regions are obtained, which not only contain text regions but also introduce many false detections. Since the difference between most characters and the background is very large, a coarse filter can be designed according to the simple morphological characteristics of MSER and the stroke characteristics of characters, which can quickly and effectively filter out a large number of false detections.

As a supplement to the preliminary coarse filtering based on simple morphological features, this paper additionally introduces features such as stroke width and coefficient of variation for further coarse filtering. For a character, the change in its stroke width is generally relatively small, but the stroke width in the background area will have a larger change. A character can be approximated by sliding a brush with the width of its stroke along its skeleton. Therefore, we can approximate the stroke width and coefficient of variation by extracting the skeleton of the character candidate region.

The calculation formulas for approximate stroke width w, stroke width variation coefficient z and skeleton color distance d are as follows:

$$\begin{aligned} w &= \frac{2}{n}\sum_{i=1}^{n} x_i, \\ z &= \frac{1}{n}\frac{\sum_{i=0}^{n-1}\left(w - 2x_i\right)^{1/2}}{w}, \\ d &= \frac{\sum_{i=0}^{n-1}\left(c_{\mathrm{iB}} + c_{\mathrm{iG}} + c_{\mathrm{iR}}\right)}{2(n-1)}. \end{aligned} \tag{1}$$

Among them, n represents the number of character skeleton pixels and c_{iR}, c_{iG}, and c_{iB} represent the intensity values of skeleton pixel i on the three color channels of R, G, and B, respectively. Based on experience and a large number of tests on the ICDAR2013 training set, the threshold of s is set to less than 0.25 h, where h represents the height of the character candidate area. For the R channel, its v is set to be less than 0.77, and the other channels are set to be less than 0.41. Setting appropriate thresholds for features such as stroke width and its coefficient of variation can effectively perform further coarse filtering on the remaining character candidate regions.

Since MSER is extracted under different channels and different scales, it is inevitable that many real characters or backgrounds will be repeatedly detected. Even after the abovementioned rough filtering, there will still be a large number of repeated character candidate regions will be retained. Therefore, we need to remove the repeated character candidate regions. This step will not only reduce the burden of the subsequent fine filter recognizer but also speed up the positioning. If the overlap ratio between any two character candidate regions is greater than 87%, they are considered to be repeated character candidate regions.

3.3. Fine Filtering Based on Convolutional Neural Network. In this section, we train a powerful two-recognition text/background convolutional neural network (CNN) recognizer to further filter out false detections that are difficult to remove by coarse filtering.

Through local perception, local information of the input text can be obtained. These local features will appear repeatedly in the text. For example, certain strokes and inflection points of the text may appear in different characters. Because of this principle, the features of the local area can be used in all positions of the entire picture. CNN uses this principle to use the same local connection weights in different positions of the text; that is, the connection weights between all neurons in each layer and the neurons in the previous layer are the same. This is weight sharing, which can further reduce the parameter number.

Due to local perception and weight sharing, a convolution kernel generally can only extract one feature. To solve this problem, convolution kernels with different weights are designed to extract different features in the text. For example, for the same person, the feature maps extracted by different convolution kernels are different. Some have a high response to the head, while some have a higher response to the torso.

In machine learning, the difference between the result predicted by the model and the actual label is called the cost function. There are many ways to get the minimum value of the cost function, and the most commonly used is the gradient descent method. The gradient descent method calculates the current gradient of the parameter each time, then advances the parameter a short distance in the opposite direction of the gradient, and repeats this until the final gradient is close to zero. At this time, the parameters obtained by the model generally just make the cost function take the minimum.

Using gradient descent method to learn parameters, for the entire data set, the cost function is

$$J(W,b) = 0.5\delta\sum_{i=1}^{w} W_i^2 + \frac{\sum_{i=0}^{m-1} J(W,b;x,y)}{m-1}. \tag{2}$$

Among them, the right term represents the mean square error between the predicted value of the model and the sample label, and the remaining term represents the regularization term, which is used to prevent overfitting. Among them, δ represents a compromise parameter between the two items. In order to make the cost function $J(w, b)$ obtain the minimum value, we require the partial derivative of the cost function and then use the gradient iteration method to obtain the local minimum:

$$\begin{aligned} W &= -\theta\frac{\partial}{\partial W_{ij}} J(W,b) + W_{ij}, \\ b &= -\theta\frac{\partial}{\partial b_i} J(W,b) + b_i. \end{aligned} \tag{3}$$

θ represents the learning rate (i.e., step size). The recognition effect of the nonlinear recognizer depends on a large extent on the number and quality of training samples and the distinguishing features extracted from these samples. In multitasking implicitly translated images, since the number of nontext backgrounds is much larger than the text, if the data set is designed according to this ratio, it will lead to overfitting, and if the data set is set at a one-to-one ratio, it will lead to insufficient training. To solve this problem, we set an unbalanced data set with a text/nontext ratio of one to two to train the CNN network.

The backpropagation algorithm includes two parts, forward propagation, and backward propagation. The essence of the above formula is to convert the partial derivative of the overall loss function into the sum of the partial derivatives of the single sample loss function. The backpropagation algorithm can effectively calculate the partial derivative of the loss function of a single sample. For a certain sample, in the forward propagation, each neuron node will calculate the weighted sum of all its connected nodes and then use the nonlinear function to input to the next layer. This is passed backward, and finally the residual between the output result and the actual result of the sample is calculated in the output layer. Then, the chain rule is used to calculate the derivative of the residual relative to the output of the previous layer. Using back propagation can effectively reduce the complexity of operations.

Based on the trained two-recognition text/background CNN recognizers, the coarsely filtered character candidate regions can be finely filtered. Only in the channel with higher contrast can the character candidate region be extracted better. Recognition also requires better contrast. In order to ensure the accuracy of recognition, you keep the character candidate region input to the CNN second recognition model in its extraction channel. However, if the character candidate area is directly zoomed and then input into the CNN second recognizer, errors will occur in some cases. For example, for the characters "*I*" and "-," if their scale is simply scaled to 32×32, they will all become solid squares. Although they belong to completely different characters, they will all be considered as misdetected and discarded. In order to solve the abovementioned problem, the surrounding area of the character candidate area is expanded by 0.1 times its height to introduce context information, and then it is scaled, which can effectively avoid this problem. Using this carefully trained two-recognition CNN recognizer, the remaining indistinguishable background part after coarse filtering can be effectively filtered out.

3.4. String Synthesis and Filtering. Words contain higher semantic information than characters, so the characters must be combined into a string. We use morphological features and geometric position to find neighboring character candidates, then use graph models to cluster these neighboring character candidates into words or text lines, and further remove false detections. This method adds features such as stroke width and skeleton color distance. In addition, different parameters are set for some of the same features. In the graph model, the character candidate regions without adjacent characters are discarded because they are likely to be noise, which can further improve the accuracy.

While extracting the MSER, we can extract the ellipse fitting with the same standard second-order central moment as the MSER. The angle between the long axis of the ellipse fitting and the x-axis is regarded as the direction of the MSER, that is, the approximate tilt direction of the character.

For the synthesis of multidirectional character strings, the restriction rules are the same as those of the horizontal direction string synthesis, with additional character direction restriction added.

We search for adjacent character candidates according to the above rules, also use the graph model to cluster the adjacent character candidate regions, keep the clusters with more than two character candidate regions, and then use the smallest area rectangle as the final text detection box. In text detection, the direction of the character string can be obtained at the same time, that is, the angle between the long axis of the minimum area rectangle and the horizontal direction.

4. Experimental Results and Analysis

4.1. Model Learning Experiment. An effective character model is the basis for character recognition and positioning. Based on the character text samples in the Char 74k data set, this section conducts character model training and examines the convergence and execution efficiency of the algorithm in the learning process. Since the input of the learning algorithm is a collection of local features, the local feature detection and description algorithm used does not have a strong relationship with the nature of the learning algorithm.

In order to make the demonstration clearer, the mixed model is not used in this section, that is, the number of submodels in the model is set to 1, and the number of Gaussian kernels corresponding to each part is also set to 1. In the initialization process, in order to avoid the influence of the initial value of the hidden variable $P\ (H)$, the initial value of the occurrence probability of the event in the set H is set to an even distribution.

Based on the above parameter settings, the EM learning algorithm for improving the deep learning network character model is implemented in the Matlab environment. Specifically, 100 character samples are used in the learning process, and the number of local features contained in each sample is between 6 and 20 (selected). The character model is trained on a dual-core PC with a main frequency of 2.7 Ghz. The maximum number of iterations is set to 60 times, and the average training time is 212.1 seconds. In the course of the experiment, the change trend of the two indicators, the average information entropy of the vector $P\ (H)$ and the average output (likelihood value) of the model on the training samples, is examined in the iterative process. Figure 4 shows the change trend of the values of the above variables with the number of iterations during the model process of training the character "F."

Figure 4 shows that the algorithm tends to converge after about 10 iterations and reaches the exit condition after 60 iterations. Among them, the average information entropy of the probability $P\ (H)$ decreases with the increase of the number of iterations and converges to a minimum after a certain number of iterations. This phenomenon means that

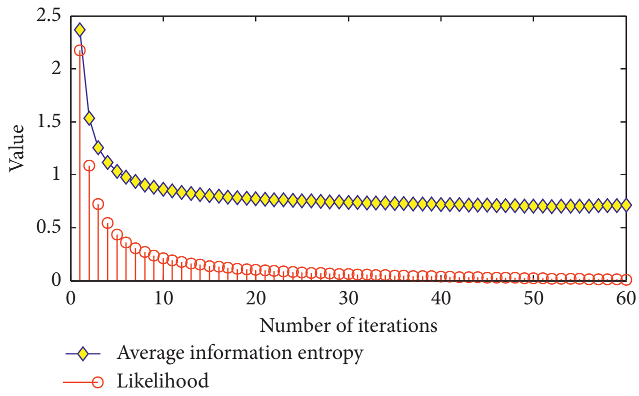


Figure 4: Trend chart of average information entropy and likelihood value with increasing number of iterations.

with the progress of the algorithm, the certainty of the hidden variable increases continuously; that is to say, the probability $P(H)$ changes from an average to a combination of one or several local features. At the same time, the likelihood value of the model on the training sample decreases with the number of iterations. When the hidden variable $P(H)$ converges, the likelihood value also converges to a minimum value. The above results prove that the learning algorithm can converge on the training samples, and the convergence is the basis for the effectiveness of model learning.

4.2. Character Detection Experiment. This section uses an improved deep learning network model for character detection in text translation. The experimental data used include ICDAR 2003 and Char 74k test set. Among them, the Char 74K data set contains English characters in the textual implication translation, including 312 textual implication translations. Figure 5 shows an example of textual translation character detection.

On the basis of the above data, we added that the data used in the final test with incomplete information included 354 English words and 1917 characters. Due to the limitation of the data, all 62 types of characters are not included in the above data. Figure 6 shows the number distribution of each type of character in the experimental sample.

Since the content of the translation contained in the text is more complex, the number of local features m obtained is often relatively large, and it is still computationally difficult to directly find a match with a larger likelihood in the text. In order to improve computational efficiency, we use a multiscale sliding window similar to PLEX + ICDAR to scan the target text in this section. In addition to filtering local features, we also use the ratio of the scale of the local feature to the window size to eliminate the local features that do not need to be considered at the current scale.

Figure 7 shows the F-score of each type of character in ICDAR 2003 and Char 74k based on the improved deep learning network character model and the random based method. The results in Figures 7(b) and 7(a) show that the performance of the method based on the improved deep learning network model on the recognition task is higher than the method based on the random synchro, and its performance in the retrieval task is better than the method based on the random synchro method.

From the experimental results, the reason why the partial-based model has better performance in retrieval tasks is that the model is not sensitive to the scale changes of the sliding window. Because the method based on randomization uses global features, the size of the sliding window is very high. In addition, characters have a strong structure, so they are easier to find in the detection process based on local saliency, and the improved deep learning network model has better handling of occlusion. These reasons all ensure that the method based on the improved deep learning network model has better results in the detection process. Correspondingly, in the recognition task, since the accurate boundary of the character has been obtained, the method based on the improved deep learning network model lacks a discriminative learning process, and the accuracy rate is low. Based on the above situation, it can be concluded that the partial-based model is more suitable for the candidate character detection process of the integrated method, and the method based on the global feature is suitable for the recognition phase of the staged method.

4.3. Character and Word Recognition Experiment. This section conducts the textual implication translation character recognition experiment, comparing the textual implication translation character recognition method based on the improved deep learning network model with several typical recognition methods, and mainly examines the discriminative ability of the model. The performance evaluation of different methods uses the average accuracy rate on all test samples.

The data used in the experiment include Char 74K and ICDAR 2003 data sets. Among them, the Char 74K data set contains a total of 7705 text samples in 62 categories, and the ICDAR 2003 data set contains 11,615 samples. The character categories include English characters (A~Z, a~z). In the above experimental data, some character classes in the ICDAR 2003 data set have insufficient samples. In this regard, artificial samples are generated by adding noise and distortion to the original samples to supplement the number of samples. During the experiment, the abovementioned character samples were segmented from the textual implication translation according to the labeling information, and each sample had a clear category label.

In the process of character model training, a mixed character model is used, and each model contains 3 submodels (experience values). Due to differences in the complexity of character classes, the number of parts n in the submodel also varies with different classes. In order to avoid the learning algorithm from falling into the trap of local extremum, randomness is introduced by randomly initializing the initial value of the hidden variable $P(H)$ in the training process.

Under the abovementioned parameter settings, the model with the best effect is selected by the verification set

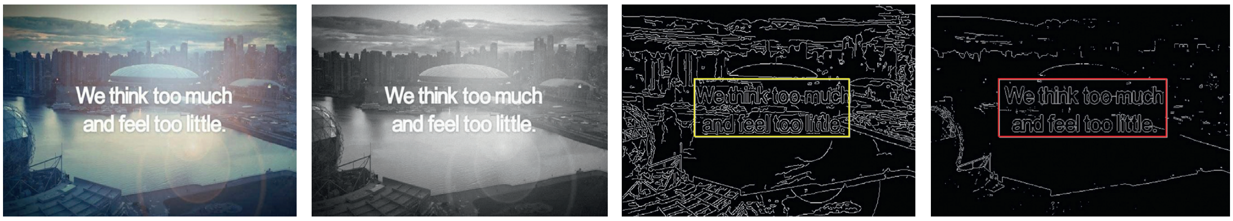


Figure 5: An example of textual translation character detection in the Char 74k dataset.

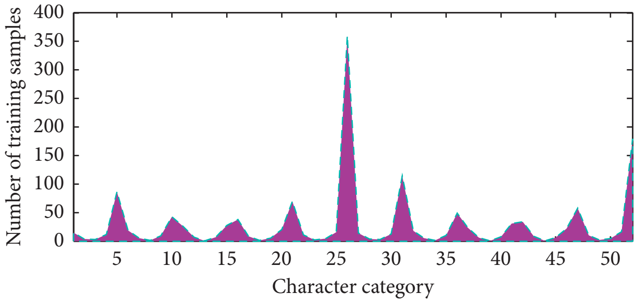
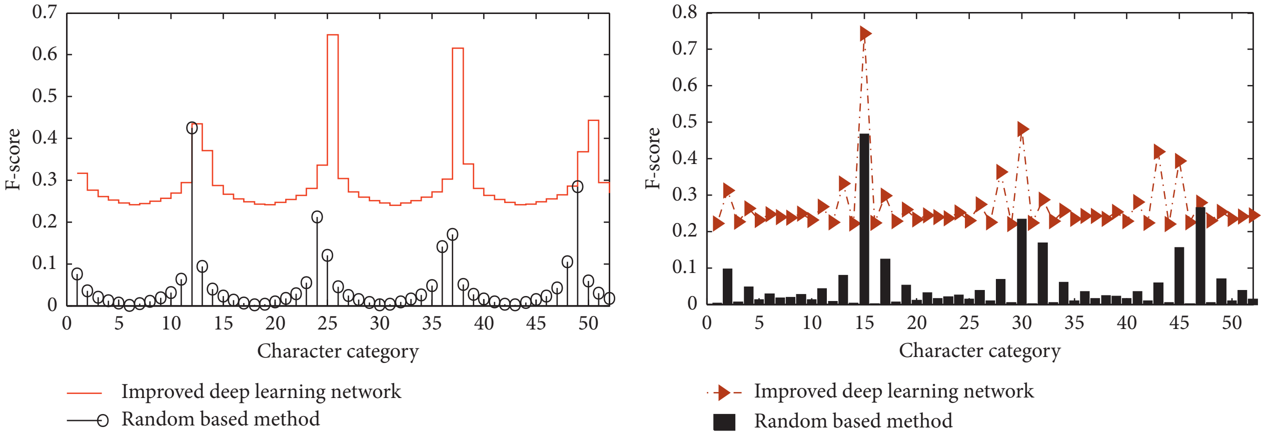


Figure 6: Distribution of the number of character classes in the sample.



(a)

(b)

Figure 7: Experimental results of character detection in textual implication translation: (a) comparison of the method based on the improved deep learning network model and the method based on random on the Char 74k test set; (b) comparison of the method based on the improved deep learning network model and the method based on random on the ICDAR 2003 test set.

for character recognition experiments. The experimental results are shown in Figure 8.

The results in Figure 8 show that the accuracy of the recognition of translated characters in text based on the improved deep learning network model is better than other methods. The main reason for this phenomenon is that the improved deep learning network is a production model and includes discriminative learning in the model learning process, so the classification ability is relatively strong. Therefore, in the recognition task of segmented isolated characters, the advantages of methods based on improved deep learning network character models are more obvious.

On the basis of isolated character recognition, we realized word recognition based on the graph structure method and compared the improved deep learning network character model with other typical methods. In the comparison process, because different methods use different enhanced information to improve the accuracy of recognition (such as the binary relationship between characters, etc.), in the experiments in this section, we do not use the above information to obtain a fair evaluation result. Among them, the predetermined vocabulary of data is obtained by learning from data samples. The results in Figure 9 show that the accuracy of the comparison method is not much different without additional information. The improved deep learning network character model performs better in word recognition tasks than the PLEX + ICDAR and HOG + RBF methods.

Comparing the result of word recognition with the result of character recognition, it can be found that the

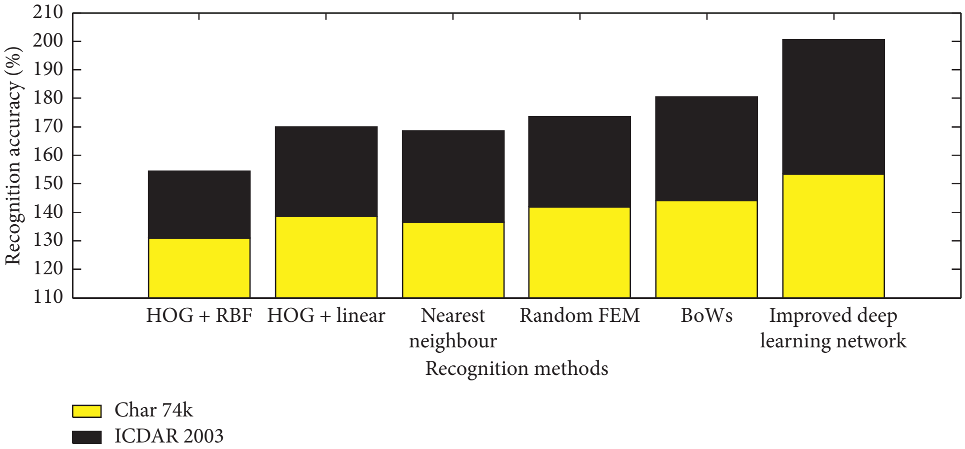


Figure 8: Experimental results of character recognition based on the improved deep learning network model.

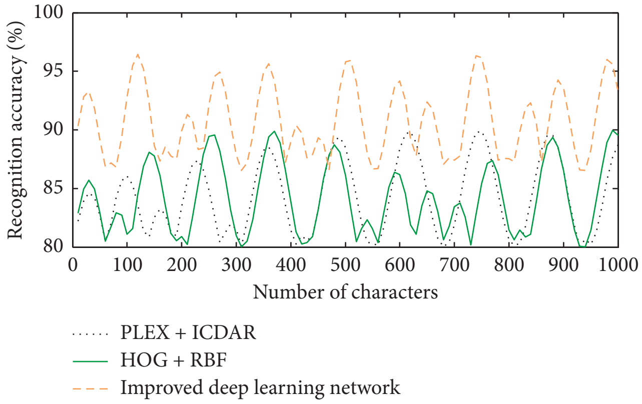


Figure 9: Experimental results of word recognition based on improved deep learning network model.

performance of the improved deep learning network model in the word recognition task is significantly better. This is mainly due to the introduction of priori knowledge of the language, which has greatly improved the overall recognition accuracy and at the same time concealed the differences between the basic character classifiers to a large extent. In addition, the performance of the improved deep learning network model is better than that of the PLEX + ICDAR method. The reason is that it is not sensitive to the scale of the sliding window and can more accurately locate the characters in the text.

5. Conclusion

This article introduces a multitask implication translation text localization method combining multichannel multiscale MSER and coarse-to-fine cascade filtering. The MSER is extracted as the character candidate area under different channels and scales, thereby effectively detecting most of the characters in the text. Using the morphological characteristics of the characters and the approximate width of the strokes and their changes for coarse filtering, a large number of false detections can be quickly removed, and then the character candidate area is removed. The CNN network fine filters the remaining character candidate regions. After cascading filtering from coarse to fine, most of the false detections can be effectively removed. The geometric position features between the remaining character candidate regions are used to find the neighboring character candidates, and the graph model is combined to fuse them into horizontal or multidirectional character strings, so as to realize the positioning of multitask translation text. In the translation of complex text implication, it is often difficult to obtain accurate results from text region detection, which leads to the performance degradation of the entire text implication translation analysis system. This paper proposes a text analysis method based on improved deep learning network character model. The model uses a collection of local features to describe the entire character and uses a probability model to model the appearance information and positional relationship of the local features and then calculate the probability of the appearance of the character. Compared with the method based on global features, the improved deep learning network character model is more flexible and can more effectively deal with the text content contained in the complex text.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Memon, M. Sami, R. A. Khan, and M. Uddin, “Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR),” *IEEE Access*, vol. 8, pp. 142642–142668, 2020.

[2] X. Chen, T. Wang, Y. Zhu, L. Jin, and C. Luo, “Adaptive embedding gate for attention-based scene text recognition,” *Neurocomputing*, vol. 381, pp. 261–271, 2020.

[3] R. Ptucha, F. Petroski Such, S. Pillai, F. Brockler, V. Singh, and P. Hutkowski, "Intelligent character recognition using fully convolutional neural networks," *Pattern Recognition*, vol. 88, pp. 604–613, 2019.

[4] L. Shao, M. Li, L. Yuan, and G. Gui, "InMAS: deep learning for designing intelligent making system," *IEEE Access*, vol. 7, pp. 51104–51111, 2019.

[5] Z. Al-Makhadmeh and A. Tolba, "Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach," *Computing*, vol. 102, no. 2, pp. 501–522, 2020.

[6] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 3, pp. 1–41, 2020.

[7] C. Luo, L. Jin, and Z. Sun, "MORAN: a multi-object rectified attention network for scene text recognition," *Pattern Recognition*, vol. 90, pp. 109–118, 2019.

[8] A. Atutxa, A. D. de Ilarraza, K. Gojenola, M. Oronoz, and O. Perez-de-Viñaspre, "Interpretable deep learning to map diagnostic texts to ICD-10 codes," *International Journal of Medical Informatics*, vol. 129, pp. 49–59, 2019.

[9] Y. Zheng, B. K. Iwana, and S. Uchida, "Mining the displacement of max-pooling for text recognition," *Pattern Recognition*, vol. 93, pp. 558–569, 2019.

[10] Q. U. A. Akram and S. Hussain, "Improving Urdu recognition using character-based artistic features of nastalique calligraphy," *IEEE Access*, vol. 7, pp. 8495–8507, 2019.

[11] M. Arsalan and A. Santra, "Character recognition in air-writing based on network of radars for human-machine interface," *IEEE Sensors Journal*, vol. 19, no. 19, pp. 8855–8864, 2019.

[12] Y. Huang, Z. Sun, L. Jin, and C. Luo, "EPAN: effective parts attention network for scene text recognition," *Neurocomputing*, vol. 376, pp. 202–213, 2020.

[13] H. El Bahi and A. Zatni, "Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26453–26481, 2019.

[14] X. Cai, S. Dong, and J. Hu, "A deep learning model incorporating part of speech and self-matching attention for named entity recognition of Chinese electronic medical records," *BMC Medical Informatics and Decision Making*, vol. 19, no. 2, pp. 101–109, 2019.

[15] K. P. Zaw and Z. M. Kyu, "Character segmentation and recognition for Myanmar warning signboard images," *International Journal of Networked and Distributed Computing*, vol. 7, no. 2, pp. 59–67, 2019.

[16] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.

[17] G. Hong, "The translation of historical documents and the study of Korean history using artificial intelligence," *International Journal of Korean History*, vol. 24, no. 2, pp. 71–98, 2019.

[18] V. S. Marco, B. Taylor, Z. Wang, and Y. Elkhatib, "Optimizing deep learning inference on embedded systems through adaptive model selection," *ACM Transactions on Embedded Computing Systems*, vol. 19, no. 1, pp. 1–28, 2020.

[19] H. Ghulam, F. Zeng, W. Li, and Y. Xiao, "Deep learning-based sentiment analysis for roman Urdu text," *Procedia Computer Science*, vol. 147, pp. 131–135, 2019.

[20] D. NguyenVan, S. Lu, S. Tian, N. Ouarti, and M. Mokhtari, "A pooling based scene text proposal technique for scene text reading in the wild," *Pattern Recognition*, vol. 87, pp. 118–129, 2019.

[21] S. Ram, S. Gupta, and B. Agarwal, "Devanagri character recognition model using deep convolution neural network," *Journal of Statistics and Management Systems*, vol. 21, no. 4, pp. 593–599, 2018.

[22] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: present contributions and future directions," *IEEE Access*, vol. 6, pp. 46019–46046, 2018.

[23] S. G. Lee, Y. Sung, Y. G. Kim et al., "Variations of AlexNet and GoogLeNet to improve Korean character recognition performance," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 205–217, 2018.

[24] D. A. Sanchez, S. G. Bulon, L. Moreno et al., "Automatic character recognition in porcelain ware," *Acta Technica Napocensis*, vol. 59, no. 3, pp. 8–12, 2018.

[25] K. Manjusha, M. Anand Kumar, and K. P. Soman, "Integrating scattering feature maps with convolutional neural networks for Malayalam handwritten character recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 187–198, 2018.