Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.020184
Article

Tech Science Press

Morphological Feature Aware Multi-CNN Model for Multilingual Text Recognition

Yujie Zhou[1], Jin Liu[1,*], Yurong Xie[1] and Y. Ken Wang[2]

[1]Shanghai Maritime University, Shanghai, 201306, China
[2]University of Pittsburgh, Pittsburgh, PA, 15260, USA
[*]Corresponding Author: Jin Liu. Email: jinliu@shmtu.edu.cn
Received: 13 May 2021; Accepted: 14 June 2021

Abstract: Text recognition is a crucial and challenging task, which aims at translating a cropped text instance image into a target string sequence. Recently, Convolutional neural networks (CNN) have been widely used in text recognition tasks as it can effectively capture semantic and structural information in text. However, most existing methods are usually based on contextual clues. If only recognize a single character, the accuracy of these approaches can be reduced. For example, it is difficult to distinguish 0 and O in the traditional CNN network because they are very similar in composition and structure. To solve this problem, we propose a novel neural network model called Morphological Feature Aware Multi-CNN Model for Multilingual Text Recognition (MFAM-CNN) in this article. We introduce a contour extraction model to enrich the representation ability, which can distinguish characters with similar shapes. Self-adaptive text density classification module is designed to recognize characters of different densities, improving the accuracy of character recognition. In general, the model is more sensitive to the overall size of the text, which improves the recognition rate of similar text. To evaluate the effectiveness of our approach, we make a dataset containing Chinese, numbers, and letters called SC dataset. Extensive experiments are conducted on the above SC dataset and ARTI-TEXT dataset, the results demonstrate that our model significantly improves the performance, achieving 98.03% and 97.77% respectively.

Keywords: Text recognition; character recognition; convolutional neural network; feature fuse; self-adaption; deep learning; cognitive computing

1 Introduction

Recent years have witnessed tremendous success in text recognition [1–2]. While commercial optical character recognition systems have reached high performance, the text feature extraction from scene images is still a challenging problem due to the wide variety of text appearance and the large number of different strokes of characters. Moreover, feature extraction is also a challenging task in both scene text and document-like text that suffers from blur or low resolution.

Most traditional methods solve this kind of problem without feature extraction. After deep learning was introduced, many methods have been proposed [3–5]. CNN has been widely used in many domains such as sentiment analysis [6], HEVC [7–9], and most classification and recognition tasks [10–14]. Such a deep network learns to extract local features, compress them to high dimensional features, and minimize the loss which represents the rate of approximate quality. These characteristics help CNN to make a better performance in recognition missions than any other models. While current text recognition methods and OCR systems [15–16] have reached good performances for simple document images, the recognition of scene images is still a challenging problem. These images often contain the multilingual character to appear at the same time. For example, Chinese characters, English, and numbers tend to appear at the same time. In this case, it is difficult to extract features from characters of different types of language in a single model. In addition, as different types of language texts increase, the categories of text also increased, which results in poor performance of models in speed. On the other hand, multilingual characters have similar shapes, which will also affect the accuracy of the model.

To solve these problems, sophisticated models are required to combine morphological features, semantic features, and cognitive computing into a complete system. In order to reduce the complexity of the system, Théodore Bluche et al. [17] and Wassim Swaileh et al. [18] share encoder and integrate multiple alphabets for multilingual recognition. These methods can only achieve satisfactory results in languages with similar alphabets. In this paper, we propose an adaptive multi-convolution neural network text with morphological features. The model is called morphological feature aware multi-CNN model for multilingual text recognition (MFAM-CNN). To differentiate the complexity of different texts, an adaptive classification algorithm which is inspired by the self-learning algorithm of cognitive computing is applied to pre-classify the text. Several different convolutional neural network models are used for recognition of different complexity texts, which can increase the overall speed of the recognition and improve the accuracy. In addition, in order to better distinguish words with similar shapes, such as "日" and "曰" in Chinese characters and "0" in numbers, "o" and "O" in English. We use the contour of the text as the input feature of our model, making the model more sensitive to the overall size of the text, thereby increasing the recognition rate of words with similar shapes. To enhance the robustness, we expanded the data set by adding multilingual characters with different morphological features.

The proposed method was tested on two different datasets. The experimental results shows that the proposed method significantly improved the performance of the text recognition. The main contributions of this study are summarized as follows:

1. A density-aware multi-CNN module has been introduced to recognize the text with different density, where the receptive field of model can be enhanced. In addition, more recognition of the complicated text with high-density could achieve a certain improvement in performance.
2. A text contour extraction module has been proposed, in which we use the contour extraction algorithm to extract the maximum bounding box of the text. Then, through the diverse aspect ratio to distinguish the text with similar appearance.

In Section 2, we describe the development status and main issues of text recognition. In addition, 4 different methods will be mentioned as recent research. Our new method MFAM-CNN will be proposed in Section 3. After that, an experimental control group and experimental results and analysis will be presented in Section 4. And in the last section, we will draw some conclusions.

2 Related Work

In this section, we review approaches that make use of recognition tasks in computer vision, with a focus on text detection and text recognition. Various methods have been proposed by researchers over the years.

At first, people's understanding of text recognition stays in the judgment of words by comparing the pixel difference between the target and the template. In 1966, Casey et al. [19] proposed the first recognition algorithm based on template matching. Such a method has been applied to other simple background text recognition methods [20–21]. This method consists of two parts, template generation, and matching algorithm. The templates contain pixel image of different characters and matching algorithm calculate the ratio of the number of errors to the correct number and then determine whether the image is the character in the template. Tsukumo et al. [22] used two different template matching methods, a nonlinear shape normalization model and a nonlinear pattern matching model, to form a handwritten Chinese character recognition model based on hierarchical classification. Such an algorithm is very effective on neatly arranged binary images of the same size.

However, this method only records the pixel information of the text and does not affect characters of different sizes, fonts, and colors. In addition, it relies on robust image pre-processing. After 2000, Support vector machine becomes a common classification tool for text recognition. The Support Vector Machine (SVM) aims to use a hyperplane to classify samples. The SVM is based on the VC dimension theory and the structural risk minimum principle in statistical learning theory. Usually, before using SVM, some feature extraction techniques are used to extract the feature from the target, and then the result is put into the SVM to classify the object. Because SVM fit nonlinear functions adaptively, it has become a very popular technology in the field of machine learning, and once achieve the art-of-state in various image classification competitions.

Gao et al. [23] proposed a Chinese character recognition model based on polynomial kernel function SVM, which significantly increased the accuracy. Shanthi et al. [24] proposed an SVM-based Teddy Milo character recognition system. In the paper, the author calculated the pixel density of 64 different regions in the image and used them as feature training SVM. Bhowmik et al. [25] proposed an SVM Bengali handwritten character recognition method based on hierarchical classification and compared SVM with multi-layer perceptron and radial basis function networks. Nasien et al. [26] proposed an SVM English handwritten character recognition model based on Free Chain Coding (FCC). The method uses free-chain coding to obtain a chain of 64 features and uses these as features in the SVM training model. Experiments show that this method has a high accuracy rate for English handwritten character recognition.

Although the support vector machine has greatly improved the recognition rate of text, its feature extraction for images still needs manual extraction to complete. To extract high-dimensional features automatically, Neural networks should be introduced which obtain image classification effects far beyond SVM.

Wu et al. [27] proposed a handwritten character recognition model (ATR-CNN) based on the relaxed convolutional neural network handwritten character recognition model (R-CNN) and the alternating training relaxed convolutional neural network. R-CNN won first place in ICDAR'13 handwritten Chinese character recognition competition [28], while ATR-CNN achieved 96.06% accuracy on the same training set. Yuan et al. [29] proposed an offline English handwritten character recognition model based on convolutional neural network. This model uses the improved LeNet-5 model to achieve 93.7% uppercase English character recognition accuracy and 90.2% lowercase English character recognition accuracy. Yang et al. [30] combined the character image itself with some specific features related to characters, such as deformation features, 8-domain features, character path features, etc., trained a handwritten Chinese character recognition model based on convolutional neural network, the method is in CASIA-OLHWDB1.0 and CASIA- OLHWDB1.1 achieved 97.20% and 96.87% accuracy, respectively. Zhong et al. [31] proposed a multi-Chinese character printed character recognition based on a convolutional neural network. Based on the existing convolutional neural network, the author adds a multi-pooling layer to the last layer. The experimental results show that this method has a good recognition rate.

In 2018, recognizing chinese characters is proposed based on time series is proposed [32]. It timings text based on columns by modeling different timings. However, there is still the problem of long and short term memory network. RNN can only record the current state and the state of the previous layer, while LSTM can control the selective memory of information by adding cell state [33].

However, for scenes where multiple characters appear at the same time, there is still no good model that can uniformly extract features and efficiently recognize them. The main reason is that it is difficult for a single model to perform uniform feature extraction for different types of language. So, improving the feature extraction part of the model is important.

Bissacco et al. [34] proposed a photo OCR system which uses over-segmentation and beam search for detection of multilingual content and uses a conditional random field for result refinement. They compared the original pixel and the HOG feature as input features, respectively, and found that using the original pixel as input makes the network deeper and wider than the original pixel when they approach the same effect. Yao et al. [35] proposed a model named strokelets. Such a model captured multiple scales structural characteristics from image characters which range from local primitives, like bar, arc, and corner of character to the whole character image. Then they divided these features into groups that contain histogram features. Ayyaz et al. [36] proposed a method named hybrid feature extraction for handwritten character recognition. The technique combined different features like structural, statical, and correlation features to a feature vector then input it into the classification module.

All these recognition methods mentioned above did not solve the problem that how to increase recognition accuracy when similar characters of different languages appear at the same time. Lyu et al. [37] took the following eight factors into consideration as a constraint to judge whether the characteristics depend on the language: orientation, stationary location, stroke density, font size, aspect ratio, and stroke statistics. And then they proposed a sliding window algorithm by two concentric squares structure. Although they have made text extraction more ubiquitous, the method they used is still the traditional method and can't be applied to multilingual text recognition models. Zhou et al. [38] proposed another multilingual text detection method which combined histogram of oriented gradient (HOG), mean of gradients (MG) and local binary patterns (LBP) as the text feature and use a cascade AdaBoost classifier to judge the probability of a text region. However, such a method focuses on text line extraction and ignores the difference of characters from different languages.

Although many methods have been proposed for text image recognition. Chinese character recognition is still a challenging problem. An important reason is that Chinese character has so large categories that general Chinese character datasets cannot be found [39]. However, instead of searching such a large scale of Chinese character data, we can try to generate it with some new technology like generative adversarial network (GAN). This network is not designed for classification as the former networks did. Its goal is to generate images which have the same pattern as the sample by giving a large amount of input. Liu et al. [40] proposed a GAN for generation of Chinese hand-written characters.

In contrast to the efforts demonstrated above, our method shows that it is possible to fuse the morphological features with the density-aware algorithm to recognize the characters.

3 Method

3.1 Overall Architecture

As illustrated in Fig. 1, we show the overall architecture of MFAM-CNN model. The input of the model is the scene image with text, which consists of Chinese characters, numbers and letters. Firstly, the contour of the text and the maximum bounding box can be obtained through the contour extraction algorithm. After that, the maximum bounding box is regarded as one input of the three different convolutional neural networks.

The automatic classification algorithm classifies the characters by adaptively calculating the overall density of the images and selects one of the convolutional neural networks to identify them. At the same time, the algorithm also includes an automatic error correction mechanism for text density misclassification. The three convolutional neural networks use different network structures and parameters based on the difference in text density. Ultimately, according to the contour ration, the position attention can be integrated into the appropriate density network to predict the final result. The algorithm and network will be introduced below. In previous research [19], we employed the flexible template matching algorithm to extract multi-scale and multi-class features of text characteristics.

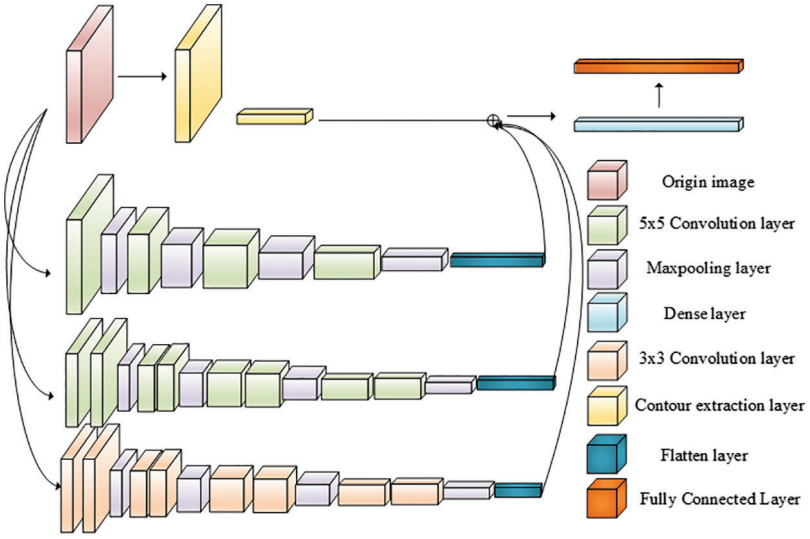

Figure 1: The architecture of the text recognition system

3.2 Text Contour Extraction

When we recognize text, we often regard the overall shape ratio of the text as an important reference standard, such as "0" in the number and "O" and "o" in the letter. These three words are similar, but compared with the letter "O", the number "0" is much slenderer. When we observe the difference in this ratio, we can easily distinguish these. At the same time, when there is noise interference in the text, the proportion of the text will not change much. We can use this ratio to make reasonable guesses on some unclear words.

Based on this, we use the contour extraction algorithm of the text to extract the maximum bounding box of the text, and a preliminary statistical contour extraction algorithm for the average aspect ratio of different types of text is shown in the Tab. 1:

Table 1: Contour extraction algorithm

//Variable Definition	
1	W: Width of Image H: Height of Image
2	x: Current abscissa y: Current ordinate
3	ulx: The abscissa of the upper left corner
4	uly: The ordinate of the upper left corner
5	brx: The abscissa of the bottom right corner
6	bry: The ordinate of the bottom right corner

(Continued)

Table 1 (continued).

```
//Variable Definition
7    ulx=−1
8    uly = −1
9    brx = −1
10   bry = −1
11   for y = 1 to H do
12      for x = 1 to W do
13         if pixel[x][y] != WHITE then
14            ulx = Min(ulx, x)
15            uly = Min(uly, y)
16            brx = Max(brx, x)
17            bry = Min(bry, y)
18      end for
19   end for
```

In this paper, we take the black body as an example and calculate the aspect ratio of 3000 Chinese characters, 10 numbers, and 52 English uppercase and lowercase letters. The statistical results are shown in the Tab. 2:

Table 2: Text aspect ratio interval statistics

Character type	Average aspect ratio
English	1.903735
Number	1.860185
Chinese	1.004891

Fig. 2 and Fig. 3 shows some of the results of the contour extraction:

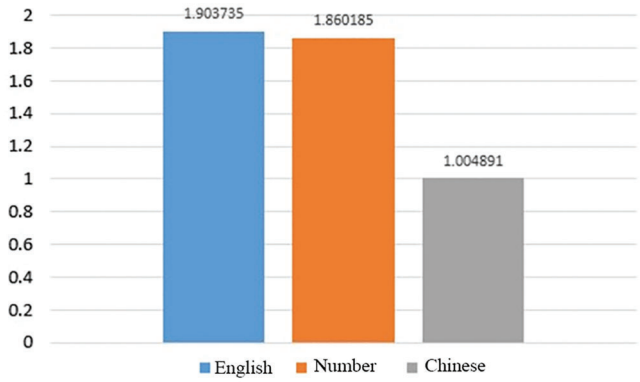

Figure 2: The average aspect ratio of different characters in the histogram

Figure 3: Results of the contour extraction

3.3 Density-Aware Multi-CNN for Text Classification

The complexity of different texts is not the same, especially for Chinese characters, there is either a single stroke such as "一" Chinese characters or a relative complex Chinese character like "鬱". For texts with relatively simple structures, using a deeper network may affect the speed of recognition. Therefore, for texts with different levels of complexity, we hope to pre-classify them with an adaptive classification method. The text is trained using neural networks of different structures to speed up recognition and increase recognition accuracy.

We assume that the maximum bounding box of the text is w × h, where M is the maximum number of pixels of the text in the maximum bounding box, and the density P of the text relative to the maximum bounding box is:

$$P = M / (w \times h) \tag{1}$$

We scale up the maximum bounding box until the larger of w and h reaches 128, and then we get M' under w' × h'. For example, if w is assumed to be the larger one, then:

$$h' = (128 \times h) / w \tag{2}$$

$$M' = (w' \times h' \times M) / (w \times h) \tag{3}$$

where w' is equal to 128, whereas if h is the larger term, a similar step is performed.

Finally, we use M' and the normalized uniform size of 128 × 128 for relative density calculations to get the final result P':

$$P' = M' / (128 \times 128) \tag{4}$$

We counted the text density of 3,000 commonly used Chinese characters, 10 numbers, and 52 English letters. The results are shown in the Fig. 4:

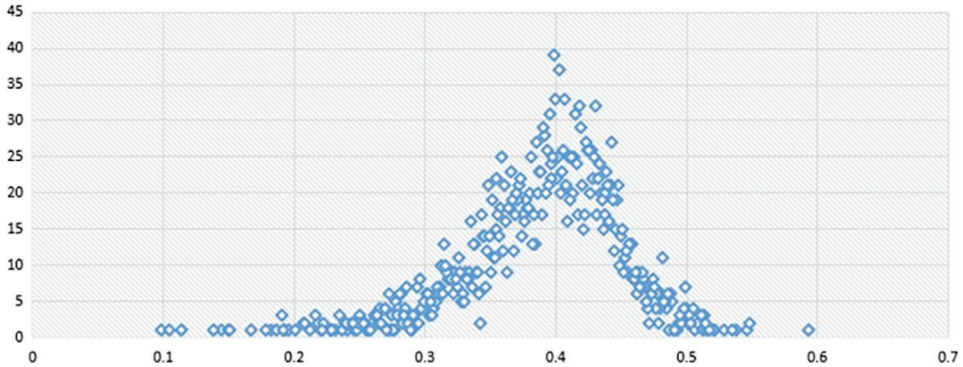

Figure 4: The distribution of the text density is depicted by the density of commonly used Chinese characters, letters and numbers statistically

It can be seen from the statistical results that most of the text aspect ratio is concentrated in the interval of 0.3 to 0.5. Therefore, we set two thresholds of th1 to 0.35 and th2 to 0.4, which can be obtained. The total numbers of words in three intervals are shown in the Tab. 3:

Table 3: The total numbers of words in three intervals

Interval	Quantity
[0, th1]	621
[th1, th2]	997
[th2, 1]	1444

And Fig. 5 shows samples of text in these three intervals. The first line contains texts in the first interval which range from 0 to th1, the second line contains texts in an interval which ranges from th1 to th2, the third line contains texts in an interval which ranges from th1 to 1.

Figure 5: Part of sample text which aspect ratio distributed in three intervals

In addition, to prevent some words at the edge of the threshold from being affected by noise and being misclassified into another class, when the recognition probability of the text in the recognition model of the current threshold is lower than a certain threshold T, we will run another model that is closer to this ratio to identify it. If the recognition probability is higher than T, we will take this result, Otherwise, multiply the recognition probability of the first selected recognition model and the recognition probability of the second recognition model by a penalty coefficient PC and then make a decision. The specific process steps are shown in the Tab. 4:

Table 4: Auto classification algorithm

```
//Variant Definition
1   T: Recognition Probability
2   PC: Penalty Coefficient
3   P: Density of character
4   I: Character Image
5   CALCULATE_P(I): Calculate density of character P of image I, demonstrated in formula (1)-(4)
6   CHOOSE_MODEL(P): Choose the recognition model not corresponding to density of character P
7   CHOOSE_ANOTHER_MODEL(P): Choose the recognition model not corresponding to density of
    character P
8   RUN_MODEL(I): Calculate the recognition probability of image I
```

Table 4 (continued).

```
    //Algorithm
9   P = CALCULATE_P(I)
10  CHOOSE_MODEL(P)
11  RESULT = RUN_MODEL(I)
12  if RESULT < T then
13     CHOOSE_ANOTHER_MODEL(P)
14     RESULT_1 = RUN_MODEL(I)
15     if RESULT_1 < T then
16        if RESULT_1 < RESULT * PC then
17           return RESULT
18        else
19           return RESULT_1
20     else
21        return RESULT_1
22   else
23     return RESULT
```

3.4 Feature Fusion

In order to make the network better distinguish similar words in different languages, we input the contour of the text as an additional feature into the network, fuse with the image features of the text itself and train the recognition model. Among them, we set the character outline feature to F, the image of the text itself to I, and the jth convolution kernel of the ith convolutional layer to be kij, then the feature map FM1 after the convolution operation on the image I can be expressed as:

$$FM_1 = I \cdot k_{11} + I \cdot k_{12} + \cdots + I \cdot k_{1j} = I \cdot \sum_{n=1}^{j} k_{1n} \tag{5}$$

where j represents the number of convolution kernels in this layer. After the first feature map is obtained, the FM obtained in the previous layer is used as input in the remaining convolutional layer, which can be expressed as:

$$FM_i = I \cdot k_{i1} + I \cdot k_{i2} + \cdots + I \cdot k_{ij} = I \cdot \sum_{n=1}^{j} k_{in} \tag{6}$$

Finally, we will flatten the j feature maps in the FM_m the mth layer to obtain a 1 × (j × w × h) of the array D, where FM_m d D can be expressed as:

$$FM_m = FM_{m-1} \cdot \sum_{n=1}^{j} k_{(m-1)n} \tag{7}$$

$$D = \sum_{n=1}^{j} \sum_{w'=1}^{w} \sum_{h'=1}^{h} (FM_{m-1} \cdot k_{(m-1)n})_{w'h'} \tag{8}$$

What else, the feature of text contour F should be flattened too, which can be expressed as below:

$$D' = \sum_{w'=1}^{w} \sum_{h'=1}^{h} F_{w'h'} = F_{11} + F_{12} + \ldots + F_{1n} + \ldots + F_{wh} \tag{9}$$

The three convolutional neural network structures used in this paper use the above method to fuse the contour features of the text with the features of the text image.

3.5 Low-Density Text Recognition Network

As shown in the Fig. 6. We use ReLU as the activation function in all convolutional layers and fully connected layers. Due to the low text density, we use a 5 × 5 convolution kernel to increase the network's receptive field. The network has two inputs, the text image itself and the maximum bounding box of the extracted text. The specific parameters are shown in the Tab. 5:

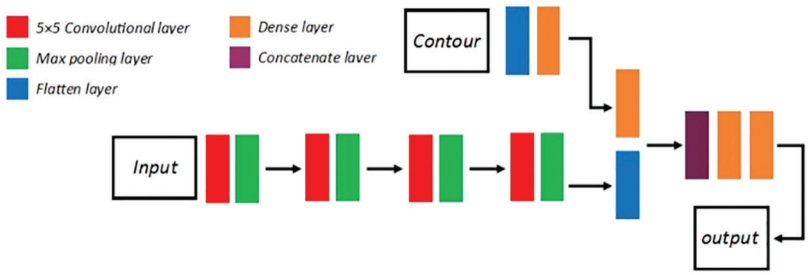

Figure 6: The architecture of the low-density text recognition network

Table 5: The input parameter of the text image

Input source	Input shape
Input	128 × 128 × 3
Contour	128 × 128 × 1

At the input of the text image, we use 4 convolutional layers and 4 pooling layers to extract the image features. The input and output parameters of the specific convolutional layer and the pooling layer are shown in the Tab. 6:

Table 6: The input parameter of the text image

Layer	Input	Output shape
Convolutional layer 1	128 × 128 × 3	128 × 128 × 16
Pooling layer 1	128 × 128 × 16	64 × 64 × 16
Convolutional layer 2	64 × 64 × 16	64 × 64 × 32
Pooling layer 2	64 × 64 × 32	32 × 32 × 32
Convolutional layer 3	32 × 32 × 32	32 × 32 × 64
Pooling layer 3	32 × 32 × 64	16 × 16 × 64
Convolutional layer 4	16 × 16 × 64	16 × 16 × 128
Pooling layer 4	16 × 16 × 128	8 × 8×128

At the contour input, the feature is extracted using an expansion layer and a fully connected layer. The specific parameters are as shown in the Tab. 7:

Table 7: Input and output parameter of extraction of the text contour

Layer	Input shape	Output shape
Flatten layer	128 × 128 × 1	16384
Fully connect layer	16384	1024

After that, we combine two features into combine layer. Two fully connected layers will be followed. The parameters are shown in the Tab. 8:

Table 8: Input and output parameter of extraction of the text contour

Layer	Input shape	Output shape
Combine layer	1024/8192	9216
Fully connect layer 1	9216	4096
Fully connect layer 2	4096	621

3.6 Middle-Density Text Recognition Networks

Fig. 7 shows the middle-density text recognition network. In this network, we still use a 5 × 5 convolution kernel to extract the text image and use ReLU as the activation function. However, unlike the low-density network, we use two layers of convolution layers to extract features from the pooled layer. At the same time, the number of convolution kernels is increased to enable the network to extract text features better. The parameters of the specific text image feature extraction process are described in Tab. 9:

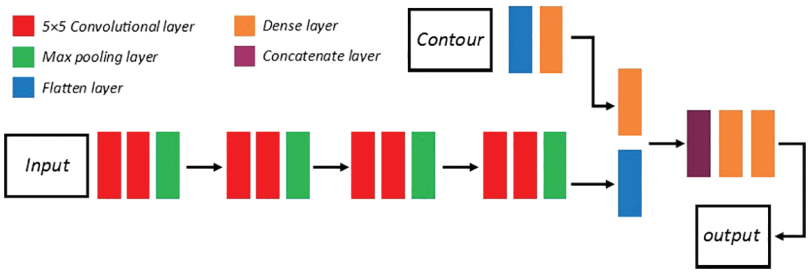

Figure 7: The architecture of the middle-density text recognition network

Table 9: Input and output parameters of the convolutional layer and pooling layer

Layer	Input	Output shape
Convolutional layer 1	128 × 128 × 3	128 × 128 × 32
Convolutional layer 2	128 × 128 × 32	128 × 128 × 32
Pooling layer 1	128 × 128 × 32	64 × 64 × 32

(Continued)

Table 9 (continued).

Layer	Input	Output shape
Convolutional layer 3	64 × 64 × 32	64 × 64 × 64
Convolutional layer 4	64 × 64 × 64	32 × 32 × 64
Pooling layer 2	64 × 64 × 64	32 × 32 × 64
Convolutional layer 5	32 × 32 × 64	32 × 32 × 128
Convolutional layer 6	32 × 32 × 128	64 × 64 × 128
Pooling layer 3	16 × 16 × 128	16 × 16 × 256
Convolutional layer 7	16 × 16 × 128	16 × 16 × 256
Convolutional layer 8	16 × 16 × 256	16 × 16 × 256
Pooling layer 4	16 × 16 × 256	8 × 8×256

At the same time, we modified the parameter of the extracted contour feature and classifier. The specific parameters are described in Tabs. 10 and 11:

Table 10: Input and output parameter of extraction of the text contour

Layer	Input shape	Output shape
Flatten layer	128 × 128 × 1	16384
Fully connect layer	16384	4096
Flatten layer	128 × 128 × 1	16384

Table 11: Input and output parameter of the classifier

Layer	Input shape	Output shape
Combine layer	4096/16384	20480
Fully connect layer 1	20480	4096
Fully connect layer 2	4096	997

3.7 High-Density Text Recognition Networks

Fig. 8 shows the high-density text recognition network. To deal with high-density text images. We use a 3 × 3 convolution kernel to extract the text image to obtain a smaller receptive field. The parameters of the feature extraction part are the same as the middle-density recognition network. The parameters of the classifier are described in Tab. 12:

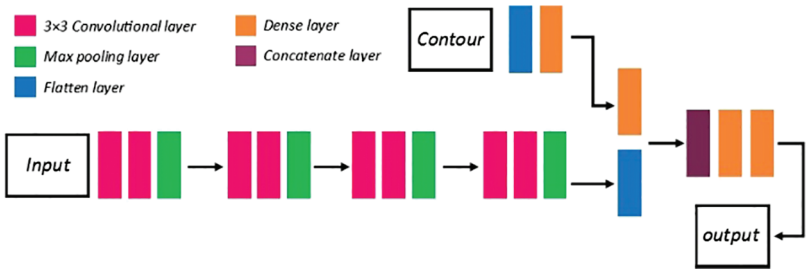

Figure 8: The architecture of the high-density text recognition network

Table 12: Input and output parameter of the classifier

Layer	Input shape	Output shape
Combine layer	4096/16384	20480
Fully connect layer 1	20480	4096
Fully connect layer 2	4096	1444

4 Experiment

4.1 Database

4.1.1 SC Dataset

In this experiment, we took 3000 black-faced Chinese prints, 52 English capitalizations, and 10 numbers as the training set for this experiment, and did a lot of morphological-based random processing on the training set, including rotation, translation, corrosion, expansion, Add random noise points, etc. Some training sets are shown in the Fig. 9.

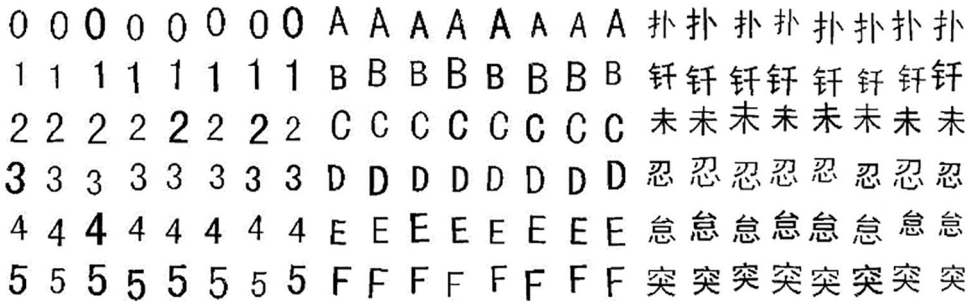

Figure 9: The black-faced Chinese sample in the dataset

4.1.2 ARTI-TEXT Dataset

In our previous work [41], we performed the operation of slicing the ARTI-TEXT dataset into characters. This article combines the previous work on the existing dataset to convert characters to text. Some examples of dataset characters are shown in the Fig. 10.

Among them, all images are 128 × 128 × 3. For each class, we randomly generated 100 training sets, of which 80 are used as training sets and 20 are used as test sets.

The training of the three networks all used Adam as the optimization function, setting the learning rate to 0.01, setting the momentum of 0.9, and the learning rate attenuation value of 10–6.

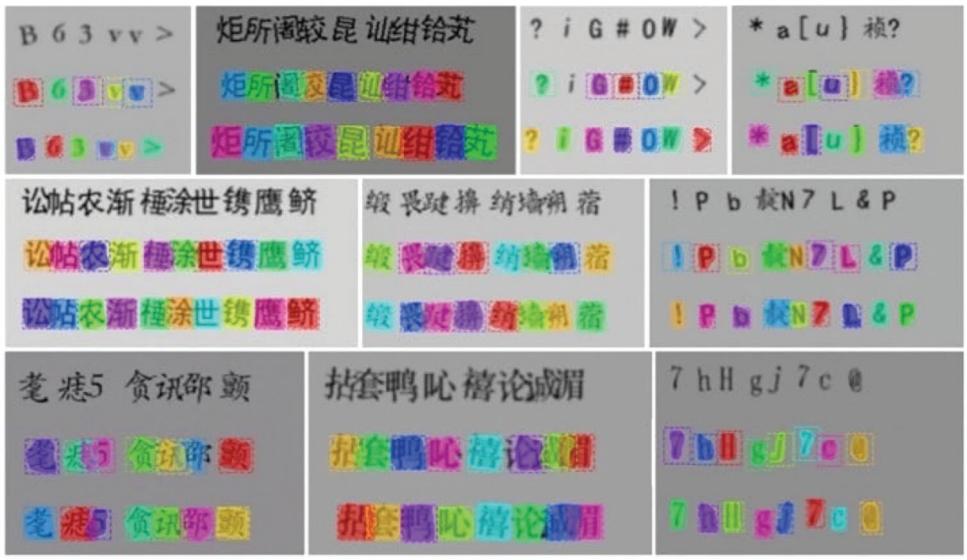

Figure 10: Sample of the ARTI-TEXT dataset

4.2 Experiment Results and Analysis

4.2.1 Experiment Result

All three networks were trained using the K80 GPU. When training the text type recognition network, we spent 5 h training the low-density network 200 rounds, spent 8 h training the medium-density network 200 rounds, and spent 12 h training 200 rounds of the high-density network, the loss function of training and the accuracy curve are as shown in Fig. 11 and Fig. 12:

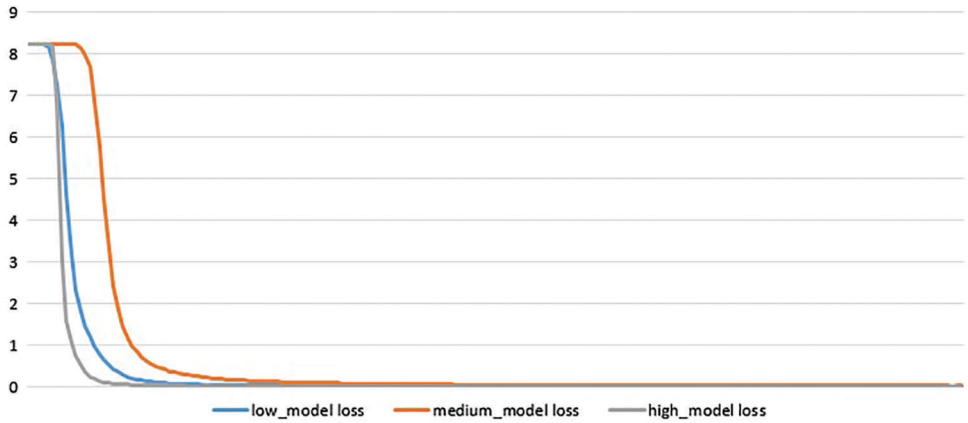

Figure 11: The loss curves

As shown in Fig. 11. and 12, they depict the loss tendency and accuracy tendency respectively. It can be seen that the faster the loss function decreases while the faster the accuracy will increase. The reason why trends change is that ¬the high-density model has fewer receptive fields and parameters, which contributes to the high speed. For instance, the middle-density model utilizes the 5×5 convolutional kernel while the high-density model employs the 3×3 convolutional kernel. As the convolutional kernel increases, the computation cost raises accordingly.

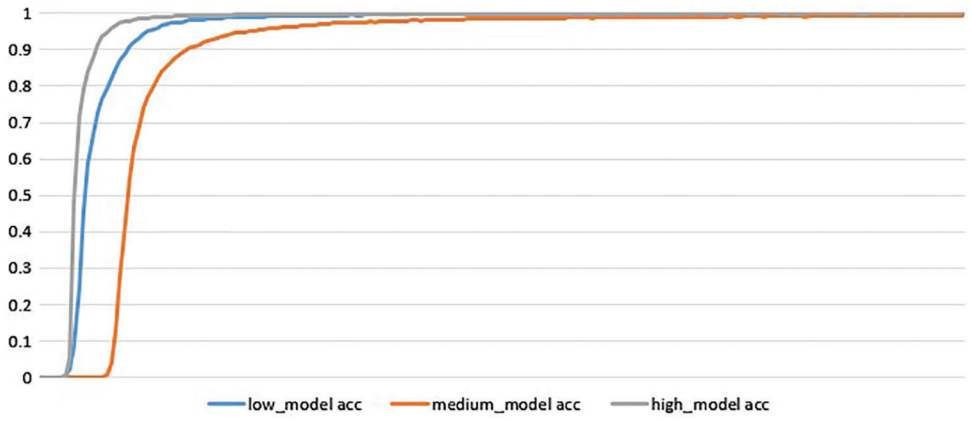

Figure 12: The accuracy curves

After training the branch models, we merged the models into a single system and tested them using three sets of training sets.

4.2.2 Experiment Comparison and Analysis

To compare and analyze the proposed model, we used the same training set to train the network structures commonly used in the CNN recognition model, such as LeNet, AlexNet, VGG, and record the accuracy of the model. All the models were trained for 200 rounds. We divided the test set into three sets to test all the trained models, and the recognition accuracy obtained is shown in the Tab. 13 and the Fig. 13:

Table 13: Contrast experimental results of the accuracy of blackbody training sets

Network	Test dataset 1	Test dataset 2	Test dataset 3
LeNet	0.9729	0.9672	0.9714
AlexNet	0.9596	0.9693	0.9653
VGG	0.9782	0.9802	0.9786
MFAM-CNN	0.9775	0.9837	0.9803

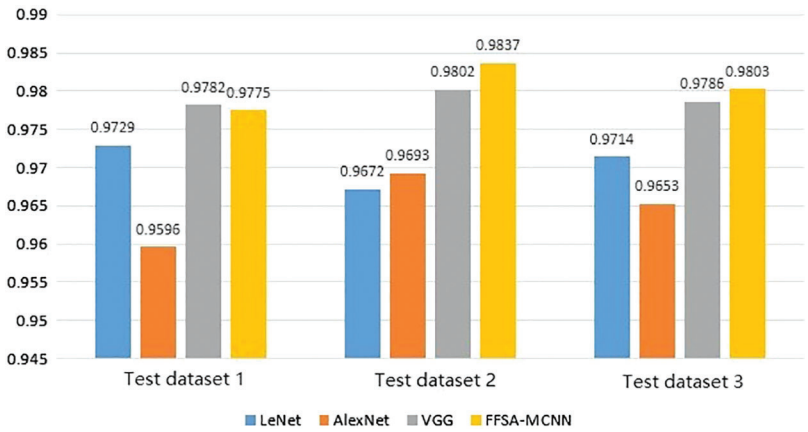

Figure 13: Contrast experimental results of the accuracy of blackbody training sets

To test the practicability of the method proposed in this chapter, we used the Chinese characters in the Chinese language format and the English uppercase and lowercase letters and numbers in the Bradley Hand ITC format to train the above four networks. All the data sets use the above self-made. A series of image processing methods used in the training set to increase the diversity of the sample. The training set of the mixed data set and the number of test sets is shown in the Tab. 14 and some of the multi-font mixed Chinese samples are shown in the Fig. 14:

Table 14: Multi-font mixed training set and the number of test sets

Dataset	English	Number	Chinese
Train set	4160	800	240000
Test set 1	520	100	30000
Test set 2	520	100	30000
Test set 3	520	100	30000

Figure 14: The multi-font mixed Chinese sample in the dataset

We use the above training set to train 4 networks and test them with 3 sets of test sets. The recognition accuracy is as shown in the Tab. 15 and the Fig. 15:

Table 15: Contrast experimental results of the accuracy of multi-font training sets

Network	Test dataset 1	Test dataset 2	Test dataset 3
LeNet	0.9422	0.9393	0.9446
AlexNet	0.9433	0.9375	0.9431
VGG	0.9483	0.9441	0.9482
MFAM-CNN	**0.9507**	**0.9487**	**0.9511**

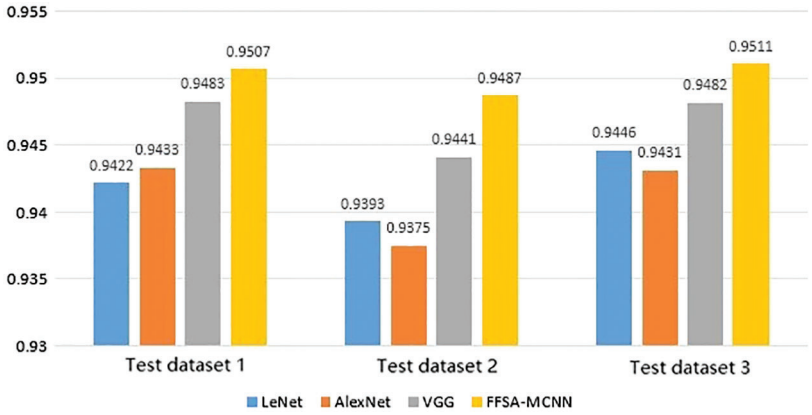

Figure 15: Contrast experimental results of the accuracy of multi-font training sets

We use the ARTI-TEXT training set to train between LeNet, AlexNet, VGG, and MFAM-CNN. The recognition accuracy is shown in the Tab. 16:

Table 16: Contrast experimental results of the accuracy of anti-text training sets

Network	Test dataset
LeNet	0.9733
AlexNet	0.9595
VGG	0.9768
MFAM-CNN	**0.9777**

5 Conclusion

In this paper, we proposed MFAM-CNN, an adaptive multi-convolutional neural network text recognition model that combines morphological features. This model fully integrates morphological features with emerging deep learning techniques. Generally, the proposed model can be used as a normal recognition model like other CNN models. According to the different lengths and widths of different types of characters, we calculated character contour for the density of characters. Then we input the character contour information into the model to increase the recognition rate of similar words. At the same time, we classify all the characters according to the density of the text and the default threshold. Different texts are identified using different network structures, which increases the efficiency of recognition and the accuracy of recognition. By comparing with other recognition models, it is proved that the proposed model has a better recognition effect on our well-tested benchmark. The model also achieves high-quality text-feature extraction from image with text with little latency.

In the future, we plan to investigate the advantages/disadvantages of the self-adaptive classification algorithm which is a central part of our approach to extend the model by adjusting the parameters or structures of our model to achieve performance comparable to state-of-the-art.

Funding Statement: This work is supported by the National Natural Science Foundation of China (61872231).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. S. Wang, S. Z. Niu and Z. G. Gao, "A novel scene text recognition method based on deep learning," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 781–794, 2019.

[2] X. Y. Wu, C. Luo, Q. Zhang, J. L. Zhou, H. Yang *et al.*, "Text detection and recognition for natural scene images using deep convolutional neural networks," *Computers, Materials & Continua*, vol. 61, no. 1, pp. 289–300, 2019.

[3] Y. Tu, Lin, J. Wang and J. U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Computers, Materials & Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[4] J. Liu, L. N. Wang, M. J. Zhou, J. Wang and S. Lee, "Fine-grained entity type classification with adaptive context," *Soft Computing*, vol. 22, no. 13, pp. 4307–4318, 2018.

[5] Y. H. Zhang, Q. Q. Wang, Y. L. Li and X. D. Wu, "Sentiment classification based on piecewise pooling convolutional neural network," *Computers, Materials & Continua*, vol. 56, no. 2, pp. 285–297, 2018.

[6] D. J. Zeng, Y. Dai, F. Li and J. Wang, "Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971–3980, 2019.

[7] L. L. Shen, X. F. Chen, Z. Q. Pan, K. F. Fan, F. Li *et al.*, "No-reference stereoscopic image quality assessment based on global and local content characteristics," *Neurocomputing*, vol. 424, pp. 132–142, 2021.

[8] Z. Q. Pan, X. K. Yi, Y. Zhang, B. Jeon and S. Kwong, "Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC," *IEEE Transactions on Image Processing*, vol. 29, pp. 5352–5366, 2020.

[9] Z. Q. Pan, X. K. Yi, Y. Zhang, H. Yuan, F. L. Wang *et al.*, "Frame-level bit allocation optimization based video content characteristics for HEVC," ACM *Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 15:1–15:20, 2020.

[10] M. Vilasini and P. Ramamoorthy, "CNN approaches for classification of Indian leaf species using smartphones," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 1445–1472, 2020.

[11] R. Y. Chen, L. L. Pan, C. Li, Y. Zhou, A. Chen *et al.*, "An improved deep fusion CNN for image recognition," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1691–1706, 2020.

[12] R. Samikannu, R. Ravi, S. Murugan and B. Diarra, "An efficient image analysis framework for the classification of glioma brain images using CNN approach," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1133–1142, 2020.

[13] J. Liu, Y. H. Yang, S. Q. Lv, J. Wang and H. Chen, "Attention-based BiGRU-cNN for Chinese question classification," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2019.

[14] S. W. Chang, J. Liu, "Multi-lane capsule network for classifying images with complex background," *IEEE Access*, vol. 8, pp. 79876–79886, 2020.

[15] S. Singh, "Optical character recognition techniques: A survey," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, no. 6, pp. 545–550, 2013.

[16] A. Chaudhuri, K. Mandaviya, P. Badelia and S. K. Ghosh, "Optical character recognition systems," *Optical Character Recognition Systems for Different Languages with Soft Computing*, vol. 352, pp. 9–41, 2017.

[17] T. Bluche and R. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," in *2017 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, pp. 646–651, 2017.

[18] W. Swaileh, Y. Soullard and T. Paquet, "A unified multilingual handwriting recognition system using multigrams sub-lexical units," *Pattern Recognition Letters*, vol. 121, pp. 68–76, 2019.

[19] R. Casey and G. Nagy, "Recognition of printed Chinese characters," *IEEE Transactions on Electronic Computers*, vol. EC-15, no. 1, pp. 91–101, 1966.

[20] J. R. Prasad, U. V. Kulkarni and R. S. Prasad, "Template matching algorithm for gujrati character recognition," in *2009 Second Int. Conf. on Emerging Trends in Engineering & Technology*, Nagpur, India, pp. 263–268, 2009.

[21] Z. J. Zhang, Y. Li and W. Q. Yuan, "Meter character recognition method based on gray template matching," in *Proc. of the 29th Chinese Control Conf.*, Beijing, China, pp. 2987–2990, 2010.

[22] J. Tsukumo, "Handprinted kanji character recognition based on flexible template matching," in *11th IAPR Int. Conf. on Pattern Recognition. Vol.II. Conf. B: Pattern Recognition Methodology and Systems*, Hague, Netherlands, pp. 483–486, 1992.

[23] X. Gao, L. W. Jin, J. Yin and J. C. Huang, "New svm-based handwritten Chinese character recognition method," *Acta Electronica Sinica*, vol. 30, pp. 651–654, 2002.

[24] N. Shanthi and K. Duraiswamy, "A novel svm-based handwritten tamil character recognition system," *Pattern Analysis and Applications*, vol. 13, no. 2, pp. 173–180, 2010.

[25] T. K. Bhowmik, P. Ghanty, A. Roy and S. K. Parui, "SVM-Based hierarchical architectures for handwritten bangla character recognition," *Document Analysis and Recognition*, vol. 12, pp. 97–108, 2009.

[26] D. Nasien, H. Haron and S. S. Yuhaniz, "Support vector machine (SVM) for English handwritten character recognition," in *2010 Second Int. Conf. on Computer Engineering and Applications*, Bali Island, Indonesia, pp. 249–252, 2010.

[27] C. Wu, W. Fan, Y. He, J. Sun and S. Naoi, "Handwritten character recognition by alternately trained relaxation convolutional neural network," in *2014 14th Int. Conf. on Frontiers in Handwriting Recognition*, Crete, Greece, pp. 291–296, 2014.

[28] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.

[29] A. Yuan, G. Bai, L. Jiao and Y. Liu, "Offline handwritten English character recognition based on convolutional neural network," in *2012 10th IAPR Int. Workshop on Document Analysis Systems*, Queenslands, TBD, Australia, pp. 125–129, 2012.

[30] W. Yang, L. Jin, Z. Xie and Z. Feng, "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge," in *2015 13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, pp. 551–555, 2015.

[31] Z. Zhong, L. Jin and Z. Feng, "Multi-font printed Chinese character recognition using multi-pooling convolutional neural network," in *2015 13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, pp. 96–100, 2015.

[32] W. Zaremba, I. Sutskever and O. Vinyals, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 40, pp. 849–862, 2018.

[33] Z. Tian, W. Huang, T. He, P. He and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *14th European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 56–72, 2016.

[34] A. Bissacco, M. Cummins, Y. Netzer and H. Neven, "PhotoOCR: reading text in uncontrolled conditions," in *2013 IEEE Int. Conf. on Computer Vision*, Sydney, Australia, pp. 785–792, 2013.

[35] C. Yao, X. Bai, B. Shi and W. Liu, "Strokelets: a learned multi-scale representation for scene text recognition," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 4042–4049, 2014.

[36] M. N. Ayyaz, I. Javed and W. Mahmood, "Handwritten character recognition using multiclass SVM classification with hybrid feature extraction," *Pakistan Journal of Engineering and Applied Sciences*, vol. 10, pp. 57–67, 2012.

[37] M. R. Lyu, J. Song and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 243–255, 2005.

[38] G. Zhou, Y. Liu, Q. Meng and Y. Zhang, "Detecting multilingual text in natural scene," in *2011 1st Int. Symp. on Access Spaces (ISAS)*, Yokohama, Japan, pp. 116–120, 2011.

[39] J. Bai, Z. Chen, B. Feng and B. Xu, "Chinese image text recognition on grayscale pixels," in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, pp. 1380–1384, 2014.

[40] J. Liu, C. K. Gu, J. Wang, G. Youn and J. Kim, "Multi-scale multi-class conditional generative adversarial network for handwritten character generation," *the Journal of Supercomputing*, vol. 75, no. 4, pp. 1922–1940, 2019.

[41] J. Zhang, J. Liu, X. Xu, P. Gong and M. Duan, "TSER: A two-stage character segmentation network with two-stream attention and edge refinement," *IEEE Access*, vol. 8, pp. 205216–205230, 2020.