ORIGINAL PAPER

Fast identification method for express end sorting label code based on convolutional recurrent neural network

Haiyan Du[1] · Chunxue Wu[1] · Yan Wu[2] · Ren Han[1] · Xiao Lin[3] · Sheng Zhang[1]

Abstract
In the automatic sorting process of express, the express end sorting label code is used to indicate that the express is dispatched to a specific address by a specific courier. Since there are many areas on the express bill containing digital information, some areas may be improperly photographed, etc. The difficulty in positioning and recognizing the express end sorting label code region is increased. To solve this problem, this paper proposes an express end sorting label code recognition method with convolutional recurrent neural network for the code specification, which has certain versatility. In order to improve the overall code recognition speed, this paper optimizes the traditional digital recognition method, removes the original segmentation operation of the character and recognizes the code as sequence recognition. Firstly, the coding region is located, and then, the express end sorting label code is recognized by the convolutional recurrent neural network. In order to test the experimental performance, this paper tests on Free-Type dataset and SUN-synthesized dataset. The experimental results show that the proposed method improves the recognition accuracy and processing speed of the express end sorting label code.

Keywords Express end · Code positioning · Label code recognition · Sequence recognition · Convolutional recurrent neural network

1 Introduction

Optical character recognition (OCR) is an important research field in the field of pattern recognition [1, 2]. It combines digital image processing, computer graphics and artificial intelligence. Digital recognition [3, 4] is also an important research direction and component of OCR. Therefore, it has attracted a large number of scholars to study digital recognition.

In the intelligent express sorting system, as long as the express end sorting label code is obtained, the address that the express is sent by the courier can be known. If the process of express sorting can be automatically processed by the computer, and the express end sorting label code is automatically extracted and accurately identified from the express bill, the time and effort for manually processing the data can be saved. Therefore, it is especially important to automatically recognize the express end sorting label code in the intelligent express sorting system.

However, the traditional digital recognition methods, such as template matching method, structural feature method and logical reasoning method, are very difficult to extract features, and the recognition effect is not very good. Until the rise of deep learning represented by convolutional neural network [5, 6] (CNN), the problem of feature extraction difficulties was solved. Therefore, a large number of scholars proposed the use of a projection method to segment a single character and then input it to CNN for character classification to achieve the purpose of recognition. But in real life, such

✉ Sheng Zhang
  zhangsheng@usst.edu.cn

  Haiyan Du
  dhy5328@126.com

  Chunxue Wu
  wcx@usst.edu.cn

  Yan Wu
  wuyan8910@126.com

  Ren Han
  campushr@163.com

  Xiao Lin
  lin6008@126.com

1 School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

2 The School of Public and Environmental Affairs, Indiana University, Bloomington 47405, USA

3 The Department of Computer Science, Shanghai Normal University, Shanghai 201400, China

as ID number, license plate number and house number, these are often a sequence. Unlike previous recognition methods, identifying these sequence objects often requires identifying a series of object labels rather than a single label. Therefore, it is necessary to regard the recognition of these objects as a sequence identification problem. Since CNN cannot handle string recognition of any length, CNN does not solve the sequence identification problem.

Another important branch of the neural network, the recurrent neural network (RNN), is mainly used to process sequence data. It only needs to convert the input image into the feature sequence of the image during preprocessing. But the preprocessing is independently of the others, so the RNN cannot be trained in an end-to-end manner. In order to construct an end-to-end system for sequence identification, some scholars have proposed combining CNN with RNN. This neural network model is named convolutional recurrent neural network (CRNN) [7, 8]. CRNN can not only perform end-to-end training, but also recognize sequences of any length.

Considering the complex background of the express bill, this paper proposes a method for first positioning and re-identifying the express end sorting label code region. In order to improve the recognition efficiency, the traditional digital recognition method is improved. The whole series of numbers is regarded as the sequence recognition problem. In this paper, the CRNN model is used. After the input image passes through the feature extraction network and the LSTM recurrent unit [9], the entire text image can be identified by the translation of connectionist temporal classification (CTC) algorithm [10]. The dataset is synthesized using SUN database images, of which 640 k images are used as training dataset and 7041 are used as test dataset, and the network model is trained on this basis. The results show that the proposed method improves the recognition accuracy and processing speed of express end sorting label code.

The rest of the paper is organized as follows. Section 2 briefly describes the work related to digital positioning and recognition. Section 3 proposes the network structure of this paper and explains each part in detail. The experimental results are presented in Sect. 4, and finally, Sect. 5 gives the summary of this paper.

2 Related works

Digital recognition is an important research direction of OCR. In this paper, express end sorting label code is regarded as a continuous sequence recognition problem, and without single character segmentation. A brief introduction to the previous works is given as follows:

1 Traditional digital recognition methods include template matching [11, 12], structural feature [13, 14] and support vector machine (SVM) [15, 16]. The template matching method is to first establish a template for each character and then use the template to compare with the character to be recognized. The identification method based on digital structure features is to use the characters to be recognized to match the structural features of the original numbers. Although this method can effectively distinguish similar characters, it is susceptible to noise. The recognition efficiency of SVM is also fine, but it is limited to the case of fewer samples, and the method requires stricter input parameters, and the selected parameters have a greater impact on the recognition. However, the development of neural networks just solves the above problems.

In 1989, Lecun et al. [17] established the modern structure of CNN and designed a five-layer CNN structure Lenet-5, which solved the recognition of handwritten numbers. However, due to the lack of training data at the time and the processing power of the computer being not particularly good, Lenet-5 was not able to solve the digital recognition in complex background. Until 2006, Krizhevsky et al. [18] proposed an eight-layer CNN structure AlexNet, which overcomes the difficulty of the original CNN to solve deep training. With the maturity of deep learning technology, deep learning technology combined with digital recognition technology has achieved good results. For example, in order to identify handwritten digit strings of unknown length, Gattal et al. [19] proposed three combinations of vertical projection, contour analysis and sliding window Radon transformation to segment the digital strings and used SVM to identify and verify each segmented digital image. In [20], handwritten Chinese text recognition was performed using neural network language model (NNLM) and CNN shape model. CNN was used for over-segmentation and geometric context modeling, and NNLM was used for character recognition. In [21], the aspect ratio detection method was used to locate the license plate area, then vertical projection method was used to segment the license plate characters, and finally, the back-propagation neural network (BPNN) was adopted to train and recognize the license plate image.

However, the character segmentation operation in the above papers is susceptible to image shadows, scratches, noise and character sticking, which may result in poor character recognition. So, some scholars have begun to research the recognition of a whole series of numbers as a sequence learning problem without dividing the characters. This method is called end-to-end recognition [22]. Even in the case of many image interference factors, it is possible to obtain a better recognition effect in the presence of noise. For example, Li et al. [23] proposed an end-to-end deep neural network that inputs license plate images into the convolutional layer to

extract regional features. The RNN for license plate recognition can share this feature. This allows the license plate to be positioned and recognized in a forward pass at the same time. Hu et al. [24] proposed a four-layer network model for end-to-end identification of Chinese text, CNN for extracting features, long short-term memory (LSTM) for processing sequences and a fully connected layer for predicting the probability of each character. CTC was used to calculate sample loss. Li. H et al. [25] proposed an end-to-end trainable text detector that can jointly detect and recognize words in natural scene images. The network consists of several convolutional layers, region proposal network, multi-layer perception, RNN encoders and RNN decoders. Wen et al. [26] proposed a deep neural network model CRNN to identify CAPTCHA. The verification code does not need any preprocessing and can be end-to-end identified, avoiding the traditional positioning, segmentation and other steps.

3 Express end sorting label code recognition system

3.1 Image collection and correction

During the actual operation of the express sorting system, the express placed on the sorting car may have a tilted picture due to the position difference, and the tilted picture needs to be corrected to improve the subsequent digital positioning and the accuracy of the recognition. Firstly, this paper collects several express images from the industrial scene. To protect the privacy of recipients and senders, their true information has been removed from all collected images. Then uses the Hough transform algorithm to correct the rotated image [27]. The result of image correction using Hough transform is shown in Fig. 1.

Figure 1a shows the original image, Fig. 1b shows the corrected pattern obtained after the HoughLines transformation is rotated by a certain angle according to the linear inclination angle.

3.2 Express end sorting label code position

The accuracy of the coding region positioning directly affects the effect of subsequent recognition. However, due to too much interference information on the express bill, it is difficult to accurately locate the express end sorting label code region in one step, so the positioning of the coding region can only be realized in a rough to fine manner. Firstly, the method in [28] is used to reduce the interference by performing a noise reduction operation on the corrected picture.

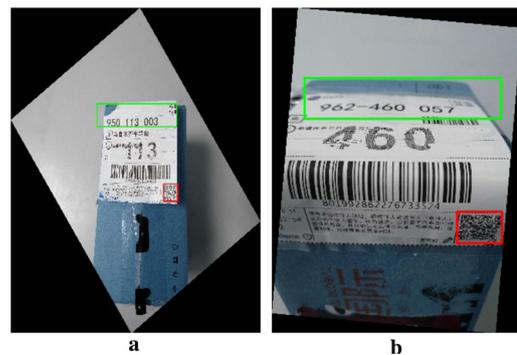**Fig. 1** Results of image correction

Fig. 2 Express end sorting label code region position

Secondly, the layout features of express bill are observed; each express picture has its corresponding QR code information. Therefore, this paper firstly uses the method in [29] to quickly detect the QR code in the image and get their respective locations in the picture. Then, the upper left corner and the lower right corner of the QR code are used as reference points. In this way, a rectangular frame can be constructed to frame the coding region to realize the positioning of the coding area. Figure 2 shows the results of positioning the coding region. The red and green bounding boxes, respectively, represent the recognition results of the QR code and the express end sorting label code.

3.3 Express end sorting label code recognition

After the coding positioning is completed, it is necessary to recognize the coding sequence. The framework of the model structure is shown in Fig. 3. Firstly, the three convolutional layers are used to extract the feature sequence of each input image. Then, each frame of the feature sequence output by CNN is predicted by RNN, and LSTM network in the RNN is used in this layer. Finally, the prediction
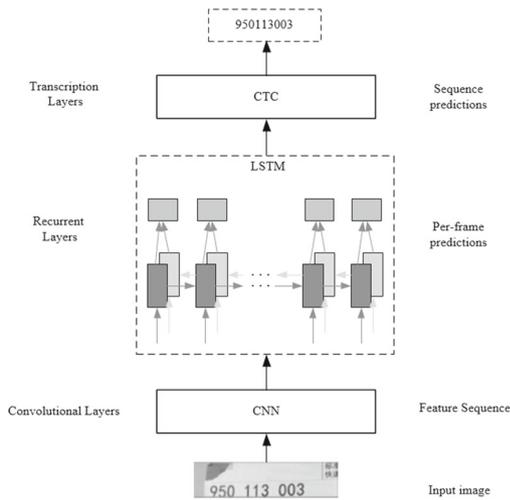
Fig. 3 Framework of CRNN

result of each frame by the RNN layer is converted into a real tag sequence by using CTC. Compared with the traditional method of first segmentation and re-identification, the performance of the entire network recognition is improved. Next, each layer of the network model is introduced in detail.

3.3.1 Convolutional layers

Sequence features are extracted from the input image using a standard convolutional neural network at the convolutional layer of the model. The structure of the convolutional layer is shown in Fig. 4. The model in this paper uses the ReLu nonlinear activation function, which replaces all negative pixels with 0. Equation of ReLu function is shown in Eq. (1):

$$f(s) = \max(0, s) \tag{1}$$

where s represents the input variable.

After the convolution and pooling operations, 256 image sequence features $S = (s_1, s_2, s_3, \ldots, s_T)$ of length 32*8 are output and the image sequence features are the input of recurrent layer. The label corresponding to each frame s_t in the image sequence feature is $L = (l_1, l_2, l_3, \ldots, l_T)$.
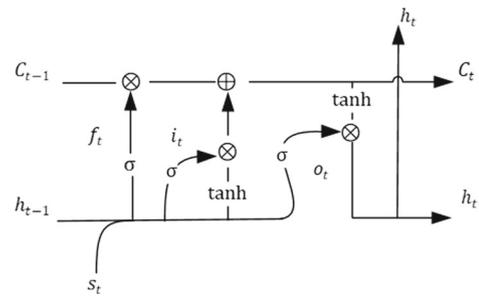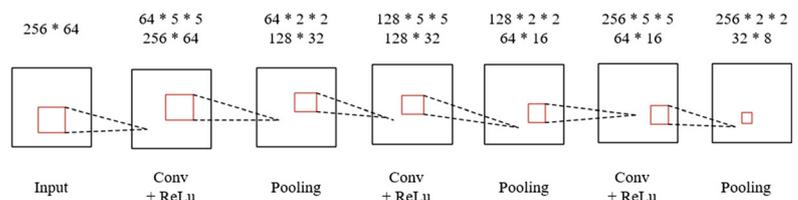

Fig. 5 Structure of LSTM

3.3.2 Recurrent layers

The input of the recurrent layer is the image sequence feature output by the convolution layer. It is to predict the label distribution l_t corresponding to each frame s_t in the image sequence feature. In this model, the recurrent unit used by RNN is LSTM shown in Fig. 5. The LSTM consists of three multiple gates, the forget gate f_t, the input gate i_t and the output gate o_t. Each gate consists of an element-level multiplication operation and a Sigmoid network layer, which can selectively add or delete information operations on the LSTM cell state C_t.

The first step of the LSTM is to filter out some of the information in the previous cell state C_{t-1} by forget gate f_t. The calculation formula of the forget gate is shown in Eq. (2):

$$f_t = \sigma\big(W_f * [h_{t-1}, s_t] + b_f\big) \tag{2}$$

where W_f and b_f represent the network weights and offsets corresponding to the forget gate, respectively. h_{t-1} represents the internal state corresponding to the current moment t and s_t represents the image feature sequence corresponding to the current time t.

The second execution step of the LSTM is to update some new information to the current cell state C_t via the input gate i_t. The calculation formula for the input gate is shown in Eqs. (3, 4, 5):

$$i_t = \sigma\big(W_i * [h_{t-1}, s_t] + b_i\big) \tag{3}$$

$$\bar{C}_t = \tanh\big(W_C * [h_{t-1}, s_t] + b_C\big) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \tag{5}$$

where $\bar{C}_t$ is a candidate value created by tanh that is selectively added to the current cell state C_t. Unwanted infor-

Fig. 4 Structure of convolutional layer

mation in the previous cell state C_{t-1} can be filtered out by $f_t * C_{t-1}$. $i_t * \bar{C}_t$ determines the extent to which new information is updated.

The third step of the LSTM is to output the implicit state h_t of the current time t through the output gate o_t. The calculation formula of the output gate is shown in Eqs. (6, 7):

$$o_t = \sigma \left(W_o * [h_{t-1}, s_t] + b_o \right) \tag{6}$$

$$h_t = o_t * \tanh(C_t). \tag{7}$$

The final step of the LSTM is to output the label distribution l_t predicted by the current time t. Its calculation formula is shown in Eq. (8):

$$l_t = \sigma (W_l * h_t + b_l). \tag{8}$$

3.3.3 Transcription layers

After a series of operations through the convolution layer and the recurrent layer, the length of the label distribution l_t may not coincide with the real label length corresponding to the image feature sequence s_t which may result in training failure. Therefore, it is necessary to use the CTC algorithm to de-integrate the label distribution predicted by the recurrent layer into the final recognition result. The CTC algorithm is specifically designed to handle the misalignment of input and output labels. For example, the label of an input image is "950113003," and after the operation of CNN and RNN, the length of the label distribution that may be output is 11. This means that a label b corresponds to a plurality of different character combinations π, and these character combinations π may be correctly translated into the label b and may also be translated incorrectly. In order to solve this problem, the CTC introduces a blank mechanism that adds a separator between successively recognized characters, denoted by #, to distinguish between the correct label sequence b and the continuously recognized label sequence. Therefore, the above input label "950113003" after the introduction of blank character combination π can have "99#5011#1#30#03," "950#111#13##00#03," etc.

After obtaining various combinations of π, it is translated into the final recognition result. Therefore, this paper adopts a mapping method $\pi = B(b)$ to remove the continuously recognized characters and separators in the output label sequence. In the case where an output label sequence is known, the formula for obtaining the sum of the probabilities of the combination π of the correct label sequence b is shown in Eq. (9):

$$p(b|L) = \sum_{\pi \in B_b^{-1}} \prod_{t=1}^{T} y_{\pi t}^t \tag{9}$$

where T represents the length of the label sequence L, y represents the probability of output of each frame in the image
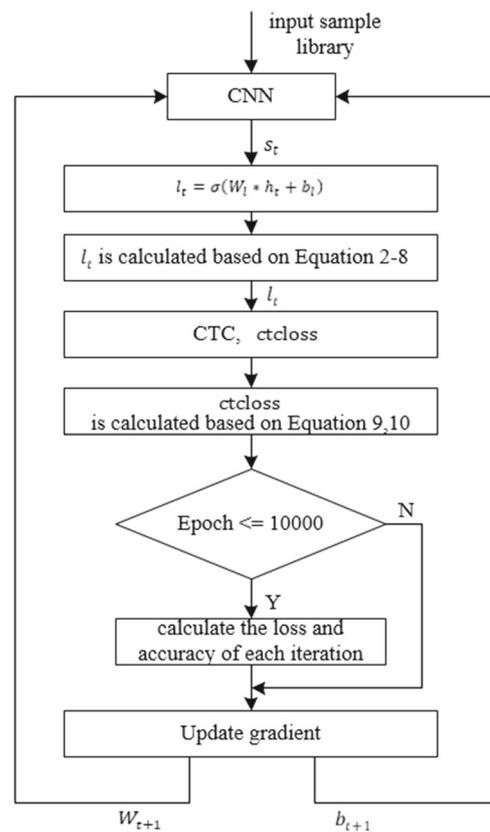

Fig. 6 Algorithm flowchart of CRNN model

feature sequence, πt represents the label combination output at time t and $y_{\pi t}^t$ represents the probability of outputting the label combination πt at time t. $p(b|L)$ also refers to the loss of a single label sequence b. What CTC has to do is to optimize this loss to a minimum using a negative maximum likelihood. The CTC loss function ctcloss is defined as shown in Eq. (10):

$$\text{ctcloss} = -\ln(p(b|L)). \tag{10}$$

3.4 Algorithm implementation

The algorithm flowchart of the CRNN model proposed in this paper is shown in Fig. 6.

Specifically, the model is divided into three parts: feature extraction network, recurrent network and CTC network. The feature extraction network uses 256*64 pictures as input, and each frame s_t in the extracted feature sequence is the output and the input of recurrent network. The output of the recurrent network is the label distribution l_t predicted by s_t at the current time. The CTC network translates l_t with the blank mechanism. In the training process of the model, the weights and offsets in the network are first randomly initialized to enable the model to perform subsequent iterative training. Then, in each subsequent iteration, each iteration is performed, and the weights and offsets generated by the

Fig. 7 The training samples of synthesized dataset

previous iteration are overwritten. The number of this paper sets the number of iterations to 10000 and saves the last generated layer weights and offsets into the data file, so that in the subsequent real-time digital recognition, the training process is omitted, and the data file is directly read to achieve the purpose of recognition and increase the speed and accuracy of recognition.

4 Experiments and results

The CRNN model proposed in this paper is developed by Python and runs on Intel Xeon machine with 2.40 GHz CPU and 32 GB RAM.

4.1 Datasets and performance

4.1.1 Datasets

The dataset used in this paper is a set of datasets synthesized by SUN database background image and digital. The size of each image is 256*64. The express end sorting label code style is strictly in accordance with the coding specifications of Xinjiang STO Express and YT Express. There are two kinds of coding forms and two different fonts; no matter which coding form is input, the recognition result is 9 digits. The training dataset contains a total of 640 k training images. They were tested on two different datasets. Some of the training images are shown in Fig. 7.

4.1.2 Performance on Free-Type dataset

In this part, 7200 images generated by the Free-Type library were used to test the recognition performance of the model in this paper. While generating a digital picture, it is possible to add a label corresponding to the content of the generated digital string to each picture, and interference factors such as noise. This article uses the Matplotlib visualization tool to display the experimental results graph. Through the experiment, the results obtained are shown in Fig. 8 and Table 1.

Figure 8a represents that when using the method in [20], the recognition accuracy tends to be stable when the model is iterated to 500 epochs, and it remains at around 95.37%. Figure 8b shows that when using the model in this paper,
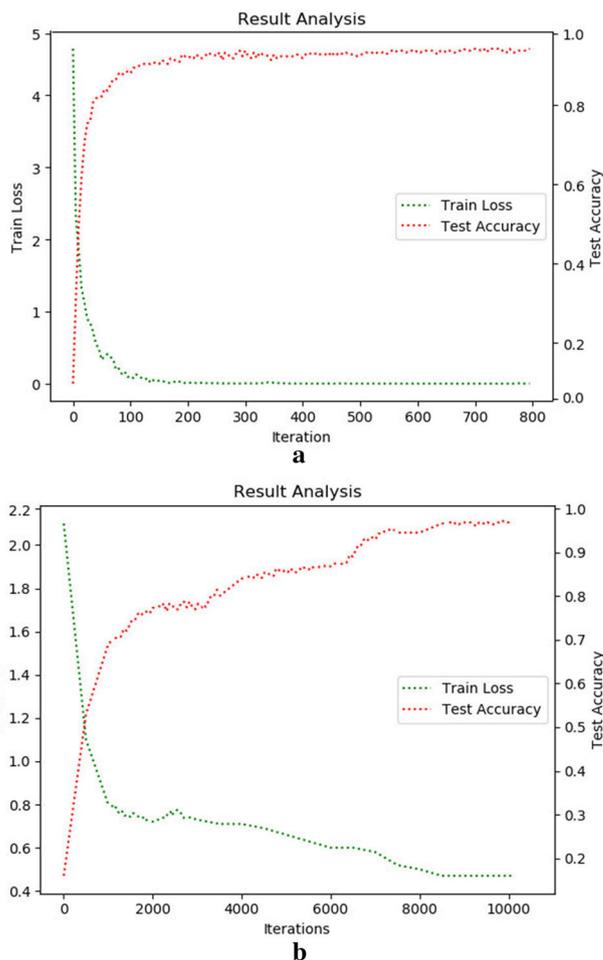

Fig. 8 Comparison plots of the two methods on Free-Type dataset

Table 1 The results based on speed and accuracy

Method	Speed (per image) (ms)	Recognition performance (%)
In [11]	1124	95.67
In [19]	1056	95.37
Ours (Free-Type)	822	96.69

the model is iterated to 8100 epochs, and the recognition accuracy reaches 96.69% and tends to be stable. As given in Fig. 8 and Table 1, the CRNN model is better for identifying the coding sequence.

4.1.3 Performance on SUN-synthesized dataset

In this part, 7041 composite images from the SUN database background image and digital were used to test the recognition performance of the model in this paper. For the method of first dividing a single character and then identifying and end-to-end recognition, the obtained result is shown in Fig. 9 and Table 2.
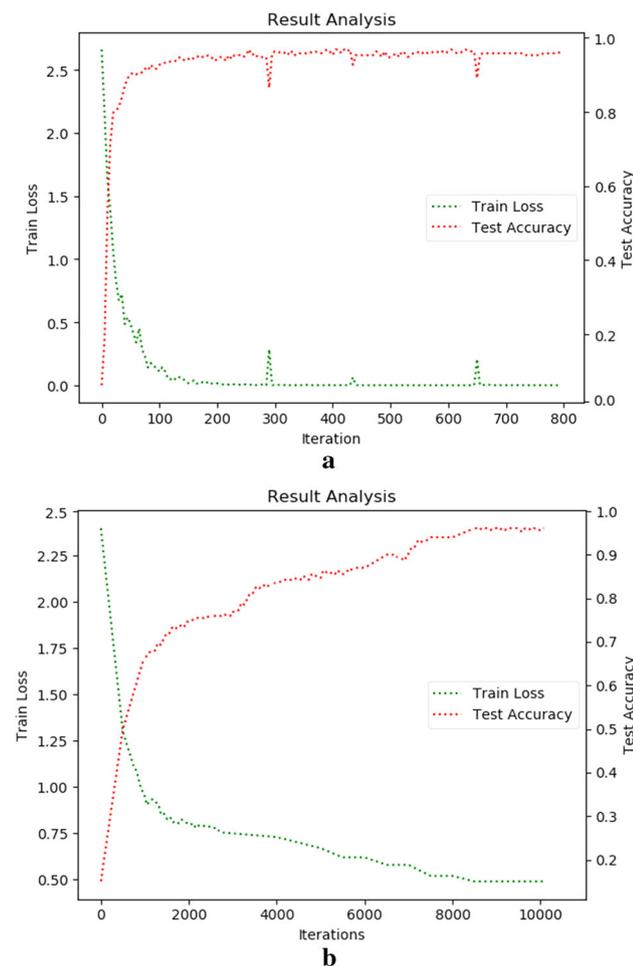
Fig. 9 Comparison plots of the two methods on SUN-synthesized dataset

Table 2 The results with different methods

Method	Speed (per image) (ms)	Recognition performance (%)
In [11]	1130	95.23
In [19]	1157	95.02
Ours (SUN)	864	96.18

#133   #267

#334   #368

Fig. 10 The result of express end sorting label code recognition

Figure 9a represents that when using the method in [20], the recognition accuracy tends to be stable when the model is iterated to 650 epochs, and it remains at around 95.26%. Figure 9b shows that when using the model in this paper, it is tested on SUN-synthesized test dataset. The model is iterated to 8260 epochs, and the recognition accuracy reaches 96.18% and tends to be stable. As given in Fig. 9 and Table 2, the method in this paper is better.

4.2 Experimental results and analysis

In this part, this paper obtains some express images from the on-site working environment of the express end sorting system. The code recognition result is displayed above the located code area. The recognition result for the express end sorting label code is shown in Fig. 10, and the results will be discussed in the following.

Figure 10 shows the recognition result of the express end sorting label code. The red and green bounding boxes, respectively, represent the recognition results of the QR code and the

express end sorting label code. In the case that the coding area is not blocked, and the line of sight is good, or the scratches are not too serious, the CRNN proposed in this paper can always correctly recognize the express end sorting label code. However, in the express bill pictures numbered 334 and 368, since the characters are partially occluded, the coding area is painted too much, and the model gives the wrong recognition result. In 334, the original "962460039" is recognized as "762460039." In 368, the original "962460016" is recognized as "962400016." But in general, the model CRNN in this paper still has better recognition performance and can handle code recognition under the conditions of rotation, light scratches and the like.

5 Conclusions and future works

This paper proposes the CRNN network model to recognize the express end sorting label code for the electronic express end sorting label code standard of two express companies in YT Express and STO Express in Xinjiang. It has certain practicability. Due to the complexity of the information on the

courier, the identification of the code uses the method of first positioning and identification. The core contribution of the proposed algorithm is to remove the character segmentation step in the traditional digital recognition method and to simplify the convolution layer and the loop layer in the model of the recognition algorithm. Therefore, the coding recognition process is simplified to a certain extent, and the influence of improper character segmentation on the code recognition is also reduced. Experiments on two datasets show that the proposed algorithm has better recognition effect on the code. But there are also certain deficiencies. In the case where the factors such as scratches and occlusion are obvious, the recognition effect is not good. In the future, image inpainting technology [30] can be used to achieve a higher recognition effect.

References

1. Caluori, U., Simon, K.: DETEXTIVE optical character recognition with pattern matching on-the-fly. Pattern Recognit. **48**(3), 827–836 (2015)
2. Sokar, G., Hemayed, P.E.E., Rehan, M.: A generic OCR using deep siamese convolution neural networks. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 1–3 Nov. 2018, pp. 1238–1244 (2018). https://doi.org/10.1109/iemcon.2018.8614784
3. Kulkarni, S.R., Rajendran, B.: Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization. Neural Netw. **103**, 118 (2018)
4. Sarkhel, R., Das, N., Saha, A.K., Nasipuri, M.: A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition. Pattern Recognition **58**(C), 172–189 (2016)
5. Rawat, W., Wang, Z.: Deep Convolutional Neural Networks for Image Classification: a Comprehensive Review. Neural Comput. **29**(9), 1 (2017)
6. Yandong, L.I., Hao, Z., Lei, H.: Survey of convolutional neural network. J. Comput. Appl. (2016)
7. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)
8. Zhang, X., Yin, F., Zhang, Y., Liu, C., Bengio, Y.: Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 849–862 (2018). https://doi.org/10.1109/TPAMI.2017.2695539
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: International Conference on Machine Learning (2006)
11. Takahashi, Y., Tanaka, H., Suzuki, A., Shio, A., Ohtsuka, S.: License plate recognition using gray image template matching with noise reduction filters and character alignment. Syst. Comput. Jpn. **38**(3), 49–61 (2010)
12. Ryan, M., Hanafiah, N.: An examination of character recognition on ID card using template matching approach. Procedia Comput. Sci. **59**(10), 520–529 (2015)
13. Vithlani, P., Kumbharana, C.K.K.C.K.: Structural and statistical feature extraction methods for character and digit recognition. J. Nucl. Mater. **127**(s 2–3), 146–152 (2015)
14. Ha, Y., Kimura, A.: Effect of recrystallization on ion-irradiation hardening and microstructural changes in 15Cr-ODS steel. Nucl. Instrum. Methods Phys. Res. **365**, 313–318 (2015)
15. Khan, M.A., Sharif, M., Javed, M.Y., Akram, T., Yasmin, M., Saba, T.: License number plate recognition system using entropy-based features selection approach with SVM. IET Image Proc. **12**(2), 200–209 (2018). https://doi.org/10.1049/iet-ipr.2017.0368
16. Sharma, S., Sasi, A., Cheeran, A.N.: A SVM based character recognition system. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (2017)
17. Lecun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (2014)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
19. Gattal, A., Chibani, Y., Hadjadji, B.: Segmentation and recognition system for unknown-length handwritten digit strings (2017)
20. Wu, Y.C., Yin, F., Liu, C.L.: Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognit. **65**(C), 251–264 (2016)
21. Wang, N., Zhu, X., Zhang, J.: License plate segmentation and recognition of Chinese vehicle based on BPNN. In: 2016 12th International Conference on Computational Intelligence and Security (CIS), 16–19 Dec. 2016, pp. 403–406 (2016). https://doi.org/10.1109/cis.2016.0098
22. Hochuli, A.G., Oliveira, L.S., Britto Jr., A.S., Sabourin, R.: Handwritten digit segmentation: is it still necessary? Pattern Recognit. **78**, 1–11 (2018). https://doi.org/10.1016/j.patcog.2018.01.004
23. Li, H., Wang, P., Shen, C.: Towards end-to-end car license plates detection and recognition with deep neural networks. In: IEEE Transactions on Intelligent Transportation Systems, p. 99 (2017)
24. Jie, H., Guo, T., Ji, C., Zhang, C.: End-to-end Chinese text recognition. In: Signal & Information Processing (2017)
25. Hui, L., Peng, W., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5248–5256 (2017). https://doi.org/10.1109/ICCV.2017.560

26. Wen, M., Xuan, Z., Cai, M.: An end-to-end verification code identification based on depth learning. Wirel. Internet Technol. (2017)

27. Yam-Uicab, R., Lopez-Martinez, J.L., Trejo-Sanchez, J.A., Hidalgo-Silva, H., Gonzalez-Segura, S.: A fast Hough Transform algorithm for straight lines detection in an image using GPU parallel computing with CUDA-C. J. Supercomput. **73**(1), 1–20 (2017)

28. Roy, A., Singha, J., Manam, L., Laskar, R.H.: Combination of adaptive vector median filter and weighted mean filter for removal of high density impulse noise from color images. IET Image Proc. **11**(6), 352–361 (2017)

29. Li, S., Shang, J., Duan, Z., Huang, J.: Fast detection method of quick response code based on run-length coding. IET Image Proc. **12**(4), 546–551 (2018). https://doi.org/10.1049/iet-ipr.2017.0677

30. Liu, F., Li, X., Liu, D.: Research of image deblurring based on the deep neural network. In: 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), 18–20 May 2018,pp. 28–31 (2018). https://doi.org/10.1109/yac.2018.8405801