Fast and Accurate Recognition for Codes on Complex Backgrounds for Real-Life Industrial Applications

Qiaokang Liang[*], Qiao Ge[*], Wei Sun[*], Dan Zhang**, Yaonan Wang*, and Kunlin Zou*

[*]*National Engineering Laboratory for Robot Vision Perception and Control, Hunan University, Changsha 410082, China*

[**]*Department of Mechanical Engineering, York University, Toronto, ON M3J 1P3, Canada*

**Corresponding Author: kunlin@hnu.edu.cn*

Submitted: 20-05-2020
Revised: 01-08-2021
Accepted: 11-08-2021

ABSTRACT

In the food and beverage industry, the existing recognition of code characters on the surface of complex packaging usually suffers from low accuracy and low speed. This work presents an efficient and accurate inkjet code recognition system based on the combination of the deep learning and traditional image processing methods. The proposed system mainly consists of three sequential modules, i.e., the character's region extraction by modified YOLOv3-tiny network, the character processing by the traditional image processing methods such as binarization and the modified character projection segmentation, and the character recognition by a Convolutional recurrent neural network (CRNN) model based on a modified version of MobileNetV3. In this system, only a small amount of tag data has been made and an effective character data generator is designed to randomly generate different experimental data for the CRNN model training. To the best of our knowledge, this report for the first time describes that deep learning has been applied to the recognition of codes on complex background for the real-life industrial application. Experimental results have been provided to verify the accuracy and effectiveness of the proposed model, demonstrating a recognition accuracy of 0.986 and a processing speed of 100 ms per bottle in the end-to-end character recognition system.

Keywords: Convolutional recurrent neural network (CRNN); Deep learning, Inkjet code character recognition and MobileNetv3.

INTRODUCTION

In the past several decades, Deep learning, a hot spot in the field of computer vision, has achieved significant successes in speech recognition (Jeyalakshmi *et al.*, 2014), detectionn (Liang *et al.*, 2019), and segmentation (Tang *et al*. 2019, Tang *et al*. 2021). Specifically, inkjet printing process has been primarily used to spray characters such as the date of manufacture and the shelf life on the surface of food packaging. Therefore, the quality of the characters is strongly affected by the performance of the code printer and other external factors. However, clear characters could provide customers with important information about the products. In what follows, failure to detect products with defective codes in a timely manner can compromise the product quality, the corporate reputation, and even the health of the customers. However, the traditional code character is usually recognized by the human eye, which will not only significantly reduce the efficiency but also consume a lot of labor and material resources (Gui *et al*., 2019; Shirmohammadi *et al*., 2014). In this scenario, it is becoming explicit that a new automated method is demanding for efficient and quick code character recognition. With the development of machine vision and deep learning technologies in the field of image processing, optical character recognition (OCR) technology starts to be widely used in the field of automatic code inspection and recognition on product packaging, which greatly improves the accuracy and efficiency of the code recognition with reduced cost (Liang *et al*., 2019).

The OCR technology is mainly composed of two steps, detection and recognition of text, which remains challenging in natural images or complex scenes (Zhu *et al*., 2016). Distinguishing the text and non-text parts in the images is the core task of text detection, which could be reduced to a simple binary classification problem. After text detection, text recognition needs to further determine the category of the text. Therefore, it requires a more detailed classification task. However, most of the existing works take the text detection and text recognition as two separate parts in research. There are also some works that combines the previous two parts into one framework. Liao, M. et al. (2017) has improved the single shot multibox detector (SSD) framework and proposed an end-to-end "Textboxes + CRNN" recognition framework for horizontally arranged texts with textboxes and CRNN for text detection and recognition, respectively. In addition, an end-to-end text recognition method is developed (Gupta *et al*., 2016), which trains a full convolutional regression network (FCRN) and text position regression to detect text, and then implements the word classifier for text recognition. In order to deal with the natural scene text arranged in any direction, another end-to-end text recognition system has firstly employed the region proposal network (RPN) to obtain the text area, and then adopted the text classifier based on the synthetic text sample training for text recognition(Busta *et al*., 2017). For single character recognition, a modified capsule network (CapsNet) employs the Connectionist Text Proposal Network (CTPN) to locate text, and then extracts a single character using the morphological processing (Singh *et al*., 2019). Also, CTDNet and CTRNet have been designed for container text detection and recognition on complex illumination conditions (Zhang *et al*., 2019). It is of interest to mention that synthetic data is used to train CRNN for Chinese character recognition (Hu *et al*., 2017). However, none of these methods can meet the speed requirements of the industrial real-time detection and it is still challenging to realize the segmentation of a single character in a complex background.

Our datasets are collected from a beverage packaging line with complex background. The resolution of the camera is 656×490 and the middle region of the original image with the size of 320×320 has been obtained as experimental data in this paper, as shown in Figure 1. As can be seen from gray-scale images in Figure 1, there are irregular streaks with brightness and darkness on the bottle, on which the code characters are printed, and different exposures on the camera lead to different brightness. Due to the complicated background pattern, it is difficult to achieve the complete segmentation of the background by using the traditional methods in natural scene. Besides, the code character is a black dot matrix and generally there exists connection between some characters, which makes it challenging to accurately extract a single character using the projection segmentation method. However, the results of character segmentation could exert a decisive influence on character recognition. Therefore, it is the objective of this paper to develop a recognition algorithm based on complex background for beverage package code, which can achieve high recognition accuracy and speed in the industrial sites.

The main advantages and benefits of the proposed code character recognition system are highlighted as follows.

1) In the character extraction stage, we use the modified YOLOv3 structure to extract code character regions. And only a small amount of tag data is used for training, which could be completed in less than 10 minutes.

2) In the character recognition stage, a CRNN based on MobileNetV3 recognition model is proposed, which is lighter and faster.

3) An improved projection segmentation method has been proposed for the effective horizontal segmentation of line characters in complex backgrounds.

4) This paper provides a fast and efficient solution for code character recognition on complex packaging. The successful application of this recognition system could provide new insight into the future use of deep learning in industry.

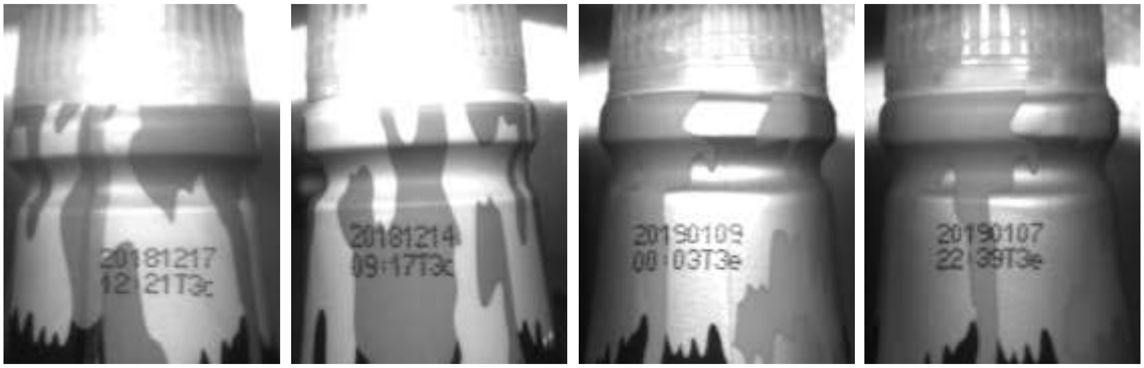

Figure 1. Examples of coding on beverage packages.

The remaining sections of this paper are prepared as follows. Section 2 briefly introduces the code character recognition system. In Section 3, the proposed methods in this system have been intensively illustrated. Typical examples also have been provided in Section 4 to demonstrate the effectiveness of the proposed system. The main conclusions and further perspectives are summarized in Section 5.

SYSTEM COMPOSITION

The code character recognition system consists of a hardware system and a software system. For the hardware system, as illustrated in Figure 2, it is mainly composed of the following components:

a) An industrial touching scree to make it convenient for operators to run the machine,
b) An image acquisition unit equipped with a Gigabit vision CCD (Baumer VLG-02C) with a resolution of 656×490 pixels and two LED light sources distributed on both sides of the detected object, which have a certain downward angle in the illumination system,
c) An embedded industrial computer equipped with Win 10 operating system and a 1050Ti graphics card, which serve as the control unit of the entire machine,
d) A lighting controller to set the lighting system modes such as the trigger mode,
e) A power unit to supply power to the system,
f) A PLC for various control functions such as switching, motion and data acquisition,
g) A heat dissipation unit to prevent machine from overheating.

The working mechanism of the hardware system can be briefly illustrated as follows. When the inspected product is transported to the designated area along with the conveyor belt, the LED light and the industrial camera will be triggered. At this time, the industrial computer will obtain images from the image acquisition system and then run the recognition program, which is directly associated with the software system. For the software system, it is compiled in python and the programs are developed by using PyQt5. The algorithm testing has been performed in the Linux system with Intel(R) Core (TM) i7-7820X CPU @ 3.60GHz, GeForce RTX 2080Ti graphics card and 11G memory. The testing environment has been configured as CUDA=10.1, Tensorflow-GPU=1.13.0, Opencv-python=4.1, and Keras=2.2.4. The implemented algorithms play a key role in the overall performance of the system and the related details will be explained in Section 3.

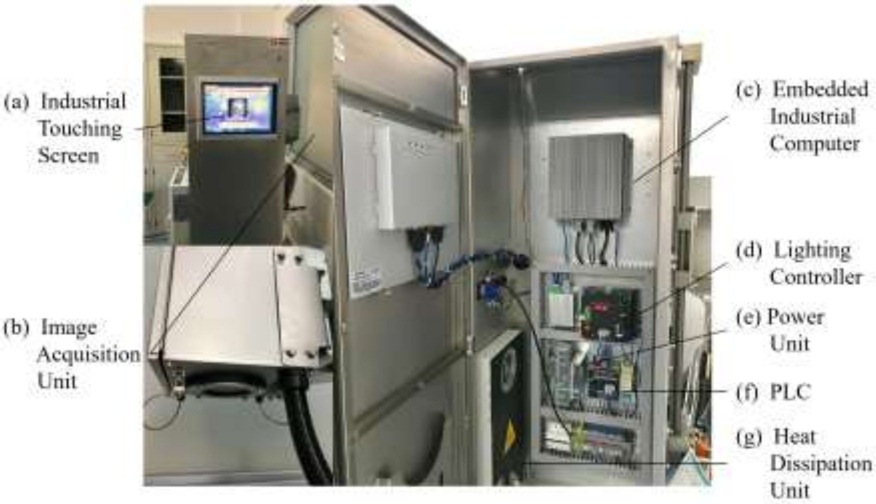
(a) Industrial Touching Screen
(b) Image Acquisition Unit
(c) Embedded Industrial Computer
(d) Lighting Controller
(e) Power Unit
(f) PLC
(g) Heat Dissipation Unit

Figure 2. The automatic code recognition system.

THE PROPOSED METHOD

The method proposed in this paper contains three modules, the character region extraction, the character processing and the character recognition. The framework of this end-to-end recognition system has been illustrated in Figure 3.

A. The character region extraction

The purpose of this module is to extract the code character region from the pictures collected from the industrial cameras as the region of interesting (ROI), the position and size of which will affect the accuracy of character recognition subsequently. The traditional algorithms of region proposal are mainly developed based on the image characteristics, such as the sliding window method based on the size information, the selective search method based on the color information(Uijlings *et al*., 2013), and the edgebox method based on edge information(Lawrence & Dollár, 2014). However, the characteristics extraction in these traditional methods are not compatible with all kinds of images. In what follows, due to the complexity of printing image background and the effects of light, it is challenging for the traditional methods to achieve accurate localization efficiently.

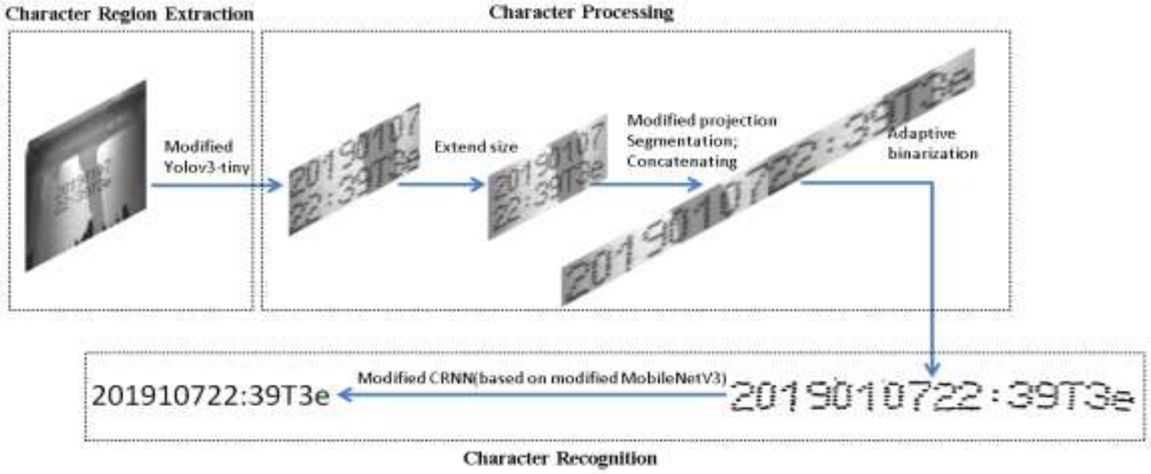
Character Region Extraction
Character Processing
Modified Yolov3-tiny
Extend size
Modified projection Segmentation; Concatenating
Adaptive binarization
201910722:39T3e
Modified CRNN(based on modified MobileNetV3)
Character Recognition

Figure 3. The proposed end-to-end character recognition method processing an image

With the significant advance of the deep learning technologies in recent years, numerous deep learning network structures have been proposed and implemented to effectively solve the problems of complex handcraft feature design and weak generalization ability existing in the traditional methods. Tian, Z., et al. (2016) has proposed the connectionist text proposal network (CTPN), which has become a typical network for text detection in OCR technology and has greatly influenced the direction of the subsequent text detection algorithms. Recently, YOLO emerges as an extremely lightweight and fast Object Detection framework. The main advantage of YOLO is utilizing the whole picture as the input of the network, which directly returns to the bounding box position and its category in the output layer.

Considering the task of single object detection in this work, this paper has selected YOLOv3-tiny, a simplified 3rd version of YOLO series(Redmon *et al*., 2016; Redmon *et al*., 2017; Redmon *et al*., 2018;). Due to its simplicity of the network and small computation, YOLOv3 has the ability to be deployed on mobile devices. Even though its accuracy is lower than that of large networks (the accuracy of bounding box localization and classification are relatively low), this method is completely competent to detect a single object in this paper. It should be mentioned that the multi-scale predictions have been removed from YOLOv3-tiny, as shown in Figure 4, and only the larger receptive field has been used to detect the object.

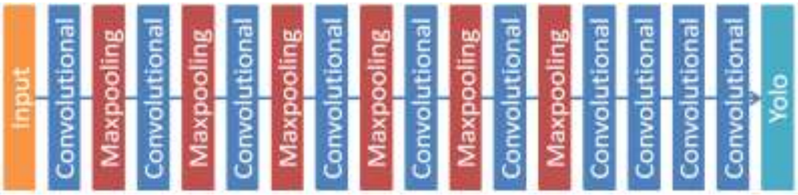

Figure 4. The architecture of the modified YOLOv3-tiny network

B. The character processing

The purpose of this module is to make preparation for the later character recognition. For the image shown in Figure 3, there are two lines in the character region with 8 characters in each line. The four parameters (x, y, w, h) of the region proposal detected by the YOLOv3-tiny network are the x coordinate of the midpoint, the y coordinate of the midpoint, the width and the height, respectively.

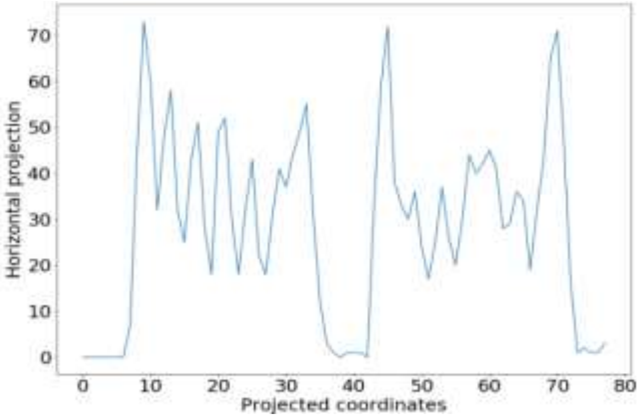

Figure 5(a). The horizontal projection of the character region

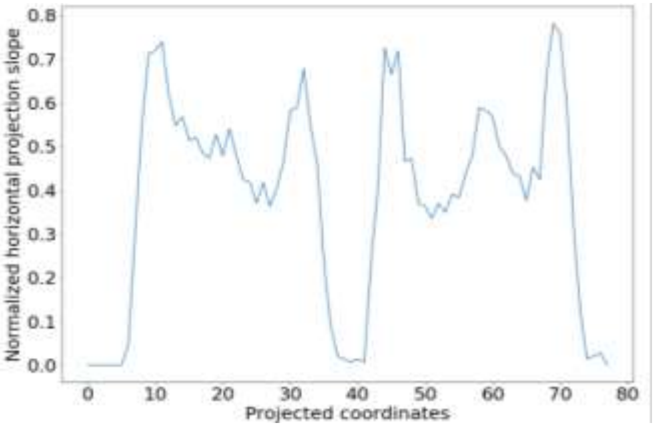

Figure 5(b). The slope of the normalized horizontal projection of the character region.

As the number of patches are smaller than expected, these patches need to be fine-tuned. Specifically, 14 pixels have been added along the length and width directions of detected patch respectively to ensure that all characters are within ROI. Afterwards, the character projection method has been utilized to cut out the redundant character blank areas. In addition, an adaptive binarization method is implemented in OpenCV to basically achieve the effect of the segmentation of the foreground and background on complex backgrounds. After binarization, obtain the y-coordinate projection by the horizontal projection method, and the corresponding result is shown in Figure 5(a). Due to the existence of the noise points in the image after binarization, and the fact that the code character is the character of the black dot matrix, it is not suitable to directly select a certain threshold to cut out images transversely. Therefore, a modified character projection segmentation algorithm is proposed in this paper. In this proposed algorithm, normalize the horizontal projection value and then get the slope of horizontal projection by

$$f\left(x\right)=\left(f\left(x-1\right)+f\left(x+1\right)\right)/2 \qquad (1)$$

With the result illustrated in Figure 5(b). The experimental results show that the range of horizontal character regions obtained by the modified method is more stable and a threshold value greater than 0.25 chosen to segment character regions has an optimal effect.

C. The character recognition

The deep convolutional neural network (CNN) has achieved great successes in various visual tasks. However, it cannot be directly applied to the prediction of the sequences of variable length, such as scene text and audio. To address this issue, a novel network structure called Convolutional Recurrent Neural Network (CRNN), which is a combination of CNN, RNN and CTC loss, has been designed specifically for recognizing sequence objects in images. In this paper, a modified CRNN model had been proposed.

1) The proposed modified CRNN model

MobileNetV3-block

MobilenetV3-block, as the main module of MobileNetV3 (Howard *et al*., 2019), integrates the key ideas of the following three models: The depth-wise separable convolutions of the MobileNetV1, the inverted residual with linear bottleneck of MobileNetV2 and a lightweight attention model based on squeeze and excitation structures.

The depth-wise separable convolution can greatly reduce the computational complexity and the number of parameters, as shown in Figure 6. The key concept of the inverted residual with linear bottleneck is inspired by the ResNet (He *et al*., 2016). Lightweight attention model has been developed as an attempt to improve the network performance by modeling the interdependencies between features and channels.

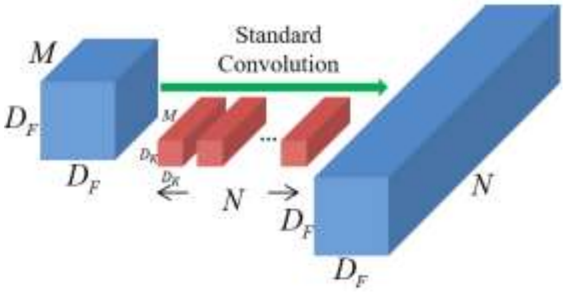

Figure 6(a). The standard convolution

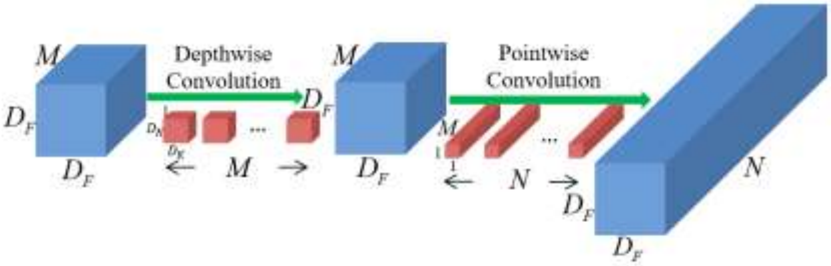

Figure 6(b). The depth-wise separable convolution

The Modified CRNN Model

The conventional CRNN models generally utilize the convolutional neural networks to extract features, then send them to a RNN in a horizontal order for sequence modeling, and finally implement a classifier to classify the features of the output at each time. The modified CRNN model in this paper, which combines the MobielNev3-block structure, adds two Maxpooling layers to extract the main features, significantly reducing the number of the parameters. In addition, the RNN part uses two layers of bidirectional GRU (Chung *et al.*, 2014) to further reduce the number of parameters and lead to rapid convergence of the model. Then, the CTC loss is developed to complete the training. The structure of the modified CRNN model has been illustrated in Figure 7, and the details of the network configuration has been shown in Table 1.

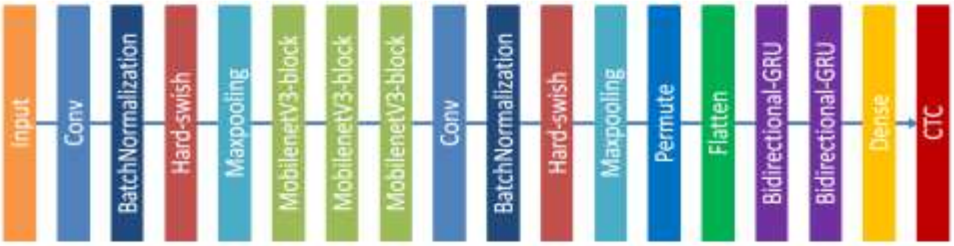

Figure 7. The structure of the modified CRNN model.

Table 1. The configuration details of the Proposed CRNN. ('k', 's', 'e' , 'bn' and 'nl' stand for kernel, stride, expansion factor, batch normalization and nonlinearity activation type, respectively).

Layer	Configuration	Output Shape
Input	-	N, 64, 720, 1
Convolutional	Maps:16, k3×3, s1×1, bn, nl='HS'	N, 64, 720, 16
MaxPooling	k2×2, s2×1	N, 32, 720, 16
MobileNetV3-block	Maps:16, k3×3, s2×2, e=16, nl='RE'	N, 16, 360, 16
MobileNetV3-block	Maps:32, k3×3, s2×2, e=72, nl='RE'	N, 8, 180, 32
MobileNetV3-block	Maps:64, k3×3, s2×2, e=120, nl='HS'	N, 4, 90, 64
Convolutional	Maps:128, k1×1, s1×1, bn, nl='HS'	N, 4, 90, 128
MaxPooling	k2×2, s2×1	N, 2, 90, 128
Map-to-Sequence	Premute + Flatten	N, 90, 256
Bidirectional-GRU	rnn_size=128	N, 90, 256
Bidirectional-GRU	rnn_size=128	N, 90, 256
CTC	-	-

2) The character data generation

To achieve the accurate code character recognition, it needs a large number of tag datasets for training. However, making labels could be very cumbersome. The real-life data is generally collected in the industrial sites in the similar period, which could result in tiny difference in the character categories and induce a great over-fitting phenomenon in the training process. In this scenario, a data generator is designed to simulate the real data production in this paper. The main procedures of the system are highlighted as follows:

(1) Select 200 original images in as many different periods as possible and cut all the single characters as templates with the template library shown in Figure 8. The characters have 63 categories including numbers, colon ":", the uppercase and lowercase letters. For each category, 10 to 30 different templates have been selected as candidates.

(2) Randomly generate 16 strings based on the date of production and the product model. Then, randomly select one from each template library corresponding to each character. Afterwards, randomly zoom in or out, adopt the OpenCV adaptive binarization method (randomly set weighting coefficient C to produce the enhanced data sets), and stitch them together to get the synthetic data, which has been illustrated in Figure 9.

The purpose of utilizing random methods is to obtain more realistic data by data enhancement in each small step.

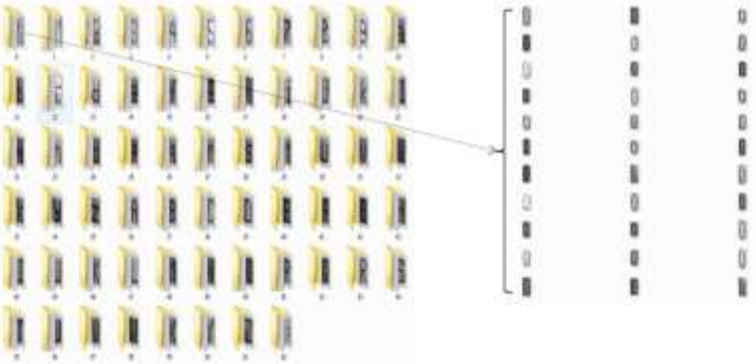

Figure 8. The library of the character templates.

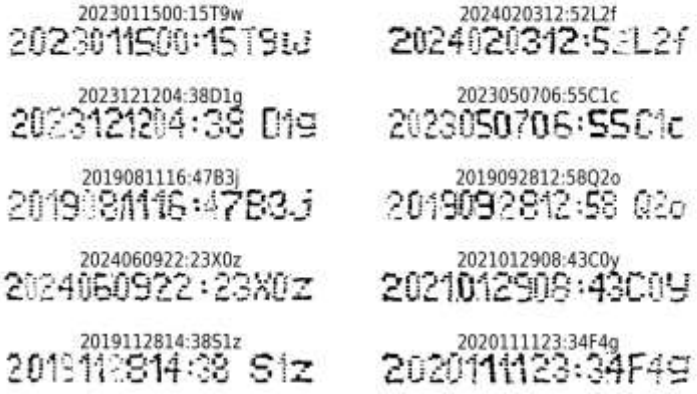

Figure 9. The sample of the synthetic data that will be used to train the proposed CRNN model.

EXPERIMENTS

In this section, experiments have been conducted to demonstrate the effectiveness of the proposed code character recognition system. In character region extraction, 1500 positive sample labels have been selected for training. To have a better convergence effect, K-means clustering method is implemented to cluster the size of region proposal to design the anchor size and transfer learning to model training with the first 15 layers of the YOLOv3-tiny pre-training model on ImageNet. In the initial stage of training, the complete YOLOv3-tiny framework has been added and the corresponding results show poor performance for small-scale prediction. Therefore, the multi-scale predictions have been removed from YOLOv3-tiny. This method only takes about 7 minutes to converge and each iteration of training consumes 266 ms, as shown in Figure 10.

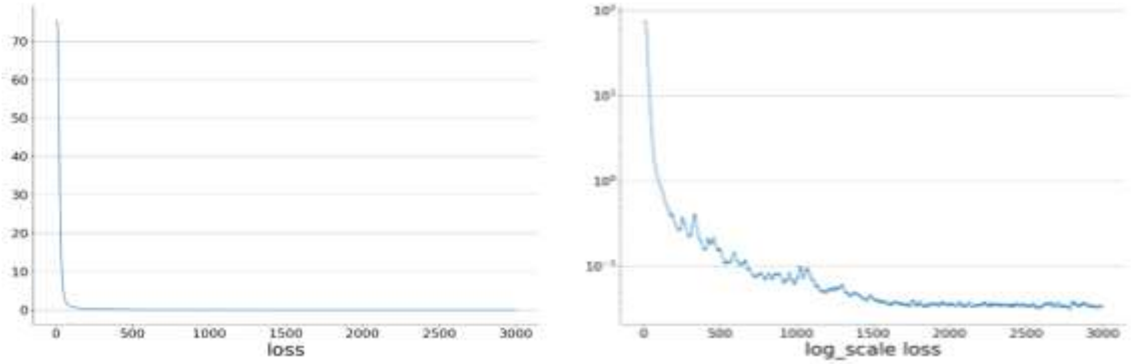

Figure 10. The loss and log scale loss of the modified YOLOv3-tiny training.

To further demonstrate the advantages of this proposed method in character region extraction, comparison has been conducted with a traditional image processing method and another deep learning method with the corresponding results shown in Figure 11. It can be clearly observed from Figure 11 that it is difficult for the traditional method to obtain a good character extraction effect on a complex background, and the deep learning method has a strong applicability. It is of interest to mention that both CTPN and the proposed method have good effects on character extraction, but the proposed method needs relatively less processing time.

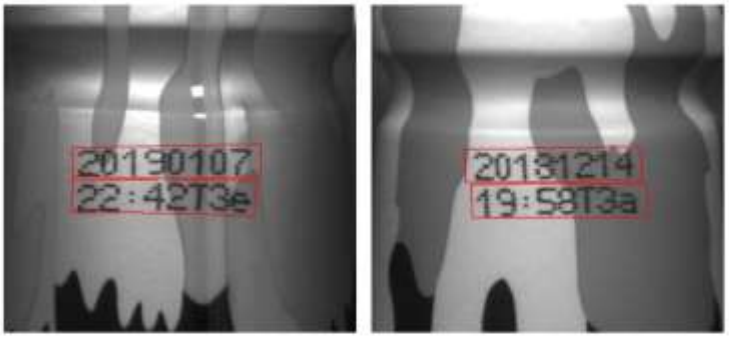

Figure 11(a). CTPN/Speed:690ms

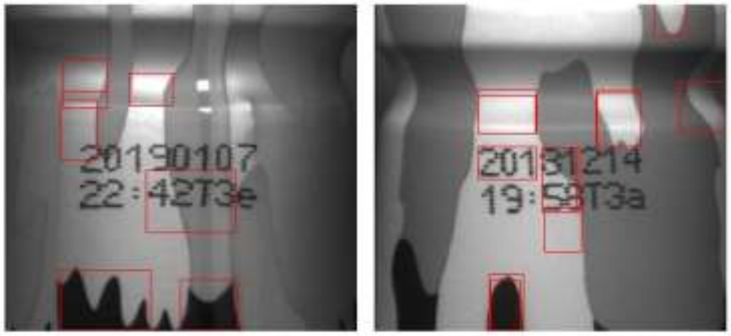

Figure 11(b). SelectiveSearch/Speed:380ms

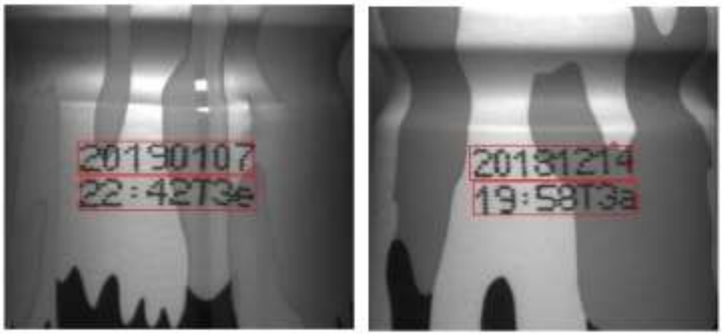

Figure 11(c). Proposed method (Character Region Extraction + Character Processing) /Speed:6.6ms

Figure 11. The comparison of three different character extraction methods.

There exist a small number of defective negative samples in the real scene with issues of character missing, ghosting, blurring, or complete missing, and so on. The experiments demonstrate that the trained model has a strong generalization ability to extract some defective coded character areas as region proposal. The recall rate of the model for positive and negative samples is as high as 99.9% and detection time can be as less as 10 ms. Moreover, the characters with defects cannot be correctly identified by the recognition model (the proposed CRNN model) and they will be classified as negative samples by comparing with the correct code characters.

In the step of the character data generation, the inkjet code characters are initially believed to be printed on the package by the code printer with fixed styles. In what follows, only one character has been selected for each character template. The experimental results show that the model trained by the synthetic data can achieve an accuracy of 99.99%, but the test results based on the real-life data are poor, indicating that the model is over-fitting by this method. Due to the data generation method proposed in this paper, the training has a better performance than before. Figure 12 shows that the loss in the training process drops quickly as the iteration number increases. As expected, the proposed model converges rapidly, which indicates that the proposed CRNN model is trainable and effective. It should be mentioned that for each iteration, 320,000 and 64,000 synthetic data have been used for training and validation, respectively.

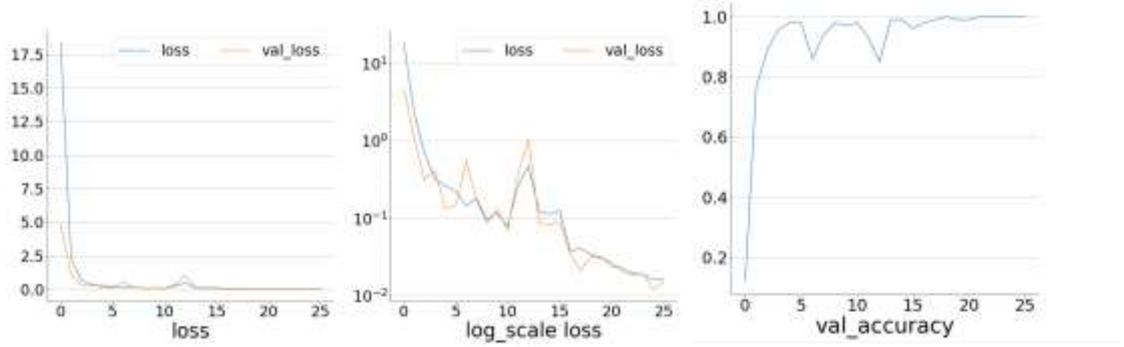

Figure 12. The loss of the training model and its accuracy on the synthetic data.

The convolutional neural network in CRNN has also employed the DenseNet structure (Huang *et al.*, 2017) and the ShuffleNetV2 structure (Ma *et al.*, 2018) respectively for the comparison with the proposed model in this paper. It should be mentioned that network structure is improved to satisfy the convolution feature extraction size at the same dimensions. For validation of the proposed model, the recognition results of 200 samples (200×16 characters) have been shown in Table 2.

Table 2. The comparison between different CRNNs.

Network Structure	**Character recognition accuracy**	**The proportion of image where all characters are correctly recognized**	**The proportion of image where at most one character is incorrectly recognized**	**Model parameters (M)**	**Average recognition speed (ms)**
CRNN based on DenseNet (RNN_size=128)	98.3%	94%	97%	1.41	230
CRNN based on ShuffleNetV2 (RNN_size=128)	98.8%	94%	96.5%	2.98	270
Proposed CRNN based on MobileNetV3 (RNN_size=64)	97.6%	93.5%	97.5%	0.28	100
Proposed CRNN based on MobileNetV3 (RNN_size=128)	98.6%	92%	98%	0.68	100

As illustrated in Table 2, the model trained by the generated data has high recognition accuracy on the real data. This result indicates that the synthetic data is close enough to the real-life scene characters to avoid the cumbersome process of making sample labels and the unbalanced number of character categories in a realistic scene, providing a convenient solution for other similar products.

Comparison has been made between the proposed end-to-end character recognition method with some existing character recognition methods based on the recognition object, the implementation method, the accuracy of each method and the application scene, as illustrated in Table 3. The results shown in Table 3 clearly demonstrate that the proposed method makes improvements for recognition on complex backgrounds, which are not available in other methods. Moreover, the proposed model in this paper has fewer parameters, higher speed and accuracy, which can well meet the requirements of the applications in the industrial sites.

Table 3. The performance comparison of different methods

Method	**Recognition Object**	**Implementation Method**	**Accuracy**	**Application Scene**
Singh *et al.* (2019)	Code printed on box	CTPN & Image processing & Modified CapsNet	0.9128	Simple background image & Large spacing of characters
Zhang *et al.* (2019)	Container text	CTDRNet	0.96	Simple background & Uneven illumination

Li *et al*. (2019)	Car license plate	Jointly-trained Network (Faster-RCNN & CRNN)	0.9804 (Best performance on AOLP dataset)	Simple background & Natural scene
Hu *et al*. (2017)	Chinese text	CRNN	0.9822 (Recognition Only)	Simple background & Chinese text & Recognition only (no detection)
The proposed method	Inkjet code on beverage package	Modified YOLOv3-tiny & Image processing & Modified CRNN	0.986	Simple or complex background image

In this paper, the proposed method features three main modules, the character region extraction, the character processing and recognition. Also, an algorithm has been developed with PyQt5 to realize the end-to-end character recognition in the industrial devices. As typical examples, two images have been recognized by the proposed model, and the corresponding results are shown in Figure 13 with the recognition time labeled.

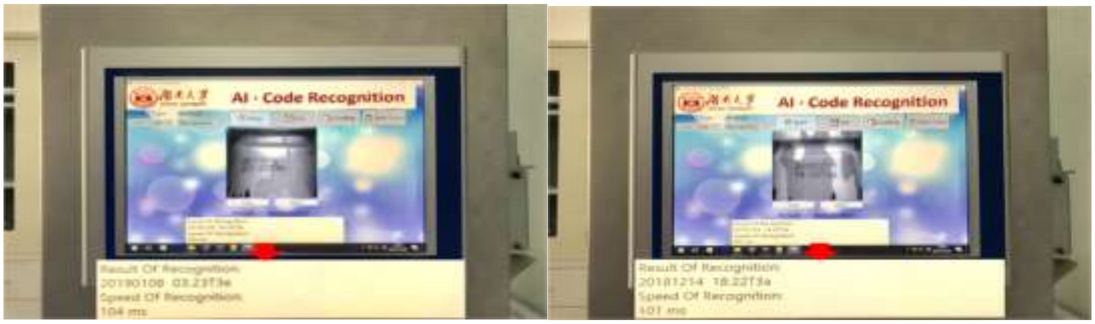

Figure 13. The character recognition application in industrial deviser.

CONCLUSION

This paper has proposed a new solution on complex backgrounds for code character recognition, combining deep learning and traditional image processing methods. Both the detection model and the recognition model adopt a lightweight network with few parameters. Also, a new data generation method has been designed in the network recognition training process, which avoids the unbalanced character categories in the training sets and the cumbersome workload in manual label making. While meeting the speed requirements, the proposed system fully exploits the powerful generalization ability of neural networks and obtains high recognition accuracy on complex backgrounds. The experimental results have verified the feasibility of this proposed system. Specifically, the character recognition accuracy can reach 0.986 with the processing speed reduced to 100 MS per bottle, which shows great potential in the practical production and applications.

The extraction of the code character region and the quality of character processing can significantly affect the accuracy of the proposed recognition model, in light of which, efforts will be made to improve the character processing method and increase the versatility of the entire algorithm for similar products and applications, in the future work.

ACKNOWLEDGEMENTS

This work was supported in part by the National Nature Science Foundation of China (NSFC 62073129), The 13th five-year plan of Educational Science in Hunan Province (Research and practice of undergraduate experimental platform for robot embedded system, XJK18BGD028), Natural Science Foundation of Hunan under Grant 2020JJ7054.

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

Busta, M., Neumann, L., & Matas, J. 2017. Deep TextSpotter: an end-to-end trainable scene text localization and recognition framework. In: Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice, Italy: IEEE, 2223-2231.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.

Gui, K., Ge, J. F., Ye, L., and Huang, L. Z. 2019. The piezoelectric road status sensor using the frequency scanning method and machine-learning algorithms. Sensors and Actuators A: Physical, 287:8–20.

Gupta, A., Vedaldi, A., & Zisserman, A. 2016. Synthetic data for text localisation in natural images. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2315-2324.

He, K. M., Zhang, X. Y., Ren, S. Q., & Sun, J. 2016. Deep residual learning for image recognition, Computer Vision and Pattern Recognition, 770-778.

Hu, J., Guo, T., Cao, J. & Zhang, C. S. 2017. End-to-end Chinese text recognition. 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, 1407-1411.

Huang, G., Liu, Z., Maaten, L. & Weinberger, K. Q.2017. Densely connected convolutional networks. Computer Vision and Pattern Recognition, IEEE, 4700-8.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. et al. 2019. Searching for MobileNetV3. arXiv preprint arXiv: 1704.04861.

Jeyalakshmi, C., Krishnamurthi, V. and Revathi, A., 2014. Development of Speech Recognition System for Hearing Impaired in Native language. Journal of Engineering Research, 2(2).

Lawrence Zitnick, C. & Dollár, P. 2014. Edge Boxes: Locating object proposals from edges [C]. European Conference on Computer Vision, 391-405.

Li, H., Wang, P., Shen, C. 2019. Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks. IEEE Transactions on Intelligent Transportation Systems. 20(3): 1126-1136.

Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D. and Sun, W., 2019. PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. Computers and electronics in agriculture, 157, pp.518-529.

Liang, Q., Zhu, W., Sun, W., Yu, Z., Wang, Y., & Zhang, D. 2019. In-line inspection solution for codes on complex backgrounds for the plastic container industry. Measurement, 148, 106965.

Liao, M. H., Shi, B. G., Bai, X., Wang, X. G., & Liu, W. Y. 2017. Textboxes: a fast text detector with a single deep neural network. In: Proceedings of the 31st AAAI Conference on Artiflcial Intelligence. San Francisco, CA, USA: AAAI, 4161-4167.

Ma, N., Zhang, X., Zheng, H. T., & Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design, in: Proceedings of the European Conference on Computer Vision (ECCV), 116–131.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You Only Look Once: Unified Real-Time Object Detection, Computer Vision and Pattern Recognition.

Redmon, J., & Farhadi, A. 2017. YOLO9000: Better Faster Stronger, Computer Vision and Pattern Recognition.

Redmon, J., & Farhadi, A. 2018. YOLOv3: An Incremental Improvement, Computer Vision and Pattern Recognition.

Shirmohammadi, S., & Ferrero, A. 2014. Camera as the instrument: the rising trend of vision based measurement. IEEE Instrumentation & Measurement Magazine, 17(3):41-47.

Singh, C. K., Gangwar, V. K., Singh, H. V., Narain, K., Majumder, A. & Kumar, S. 2019. Deep Capsule Network based Automatic Batch Code Identification Pipeline for a Real-life Industrial Application. 2019 International Joint Conference on Neural Networks (IJCNN).

Tang, P., Liang, Q., Yan, X., Xiang, S., Sun, W., Zhang, D. and Coppola, G., 2019. Efficient skin lesion segmentation using separable-Unet with stochastic weight averaging. Computer methods and programs in biomedicine, 178, pp.289-301.

Tang, P., Yan, X., Nan, Y., Xiang, S. and Liang, Q., 2021. Feature Pyramid Non-local Network with Transform Modal Ensemble Learning For Breast Tumor Segmentation in Ultrasound Images. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control.

Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. 2016. Detecting Text in Natural Image with Connectionist Text Proposal Network. European Conference on Computer Vision, 56-72.

Uijlings, J. R. R., Van de Sande, K. E. A. & Gevers, T. 2013. Selective search for object recognition [J]. International Journal of Computer Visio, 104(2), 154-171.

Zhang, W., Zhu, L., Xu, L., Zhou, J., Sun, H., and Liu, X. 2019. Deep Learning Based Container Text Recognition, 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD), 69-74.

Zhu, Y. Y., Yao, C., and Bai, X. 2016. Scene text detection and recognition: recent advances and future trends. Frontiers of Computer Science, 10(1):19-36.