Port container number recognition system based on improved YOLO and CRNN algorithm

XingQi Feng[1] *
School of Automation
Wuhan University of Technology
Wuhan 430070, Hubei, China
fxq1067544213@163.com

ZhiWei Wang[2]
School of Automation
Wuhan University of Technology
Wuhan 430070, Hubei, China
605421170@qq.com

TongCai Liu[3]
School of Automation, Wuhan University of Technology,
Wuhan 430070, Hubei, China
790341817@qq.com

***Abstract*—Port container number recognition algorithm, as an application example of OCR technology, is developing rapidly. However, the traditional container number identification algorithm from image preprocessing to location and then segmentation is not only becoming more and more time-consuming, but also has many disadvantages. At present, most of the deep learning algorithms are multi-stage, which need multi-stage adjustment in training. In order to solve the above problems, this paper proposes a container number recognition algorithm based on improved lightweight YOLOV3 and improved CRNN. That is, on the basis of the original algorithm, by reducing the number of parameters and the number of residuals in each layer of the original network, we use the deep separable convolution, introduce the Inverted Residuals block of MobileNetV2, and reorganize residual blocks to get lightweight. The end to end recognition algorithm has the advantages of high accuracy and fast speed, the accuracy of case recognition is 96%.**

Keywords—deep learning; YOLOv3; CRNN; container number recognition;

I. INTRODUCTION

The core of the container number recognition system is the optical character recognition OCR [1], that is, the process of detecting and recognizing characters in the image by simulating human processing methods. OCR can be mainly divided into three parts: preprocessing, text detection and text recognition.

In the field of text detection, there are mainly methods based on image processing, such as object contour extraction, feature based object contour extraction and deep learning target detection. In the early text detection field, features were manually designed to capture the scene text features. For example, Zhuo Junfei[2] and other scholars proposed using the edge feature and vertical projection method to locate the target area. However, due to the detection area pollution, environmental shadow and other factors, the target location is inaccurate. Later, Wan Yan and others put forward another color feature detection method[3], using coarse texture information to detect the target candidate area, and then used different RGB values of three channels to cluster different colors of the same color. The target area is accurately positioned by means of connectivity analysis. However, the gray value of color is greatly influenced by weather and light, so positioning in practice is not reliable.

Traditional methods are easily influenced by image quality and environment, while deep learning methods have become mainstream, but most algorithms such as R-CNN[4], fast-RCNN and so on are mostly algorithms. It is multi-stage, and the two phases of candidate area and object recognition are relatively independent. They contain multiple stages of candidate extraction and candidate frame filtering, which may be suboptimal and time-consuming.

In recent years, one-stage detection algorithms have emerged, such as YOLO9000[5] and YOLOv3[6]. These algorithms directly generate the location and class information of the target directly through the network. It is an end-to-end target detection algorithm with high speed and high accuracy. Therefore, the one-stage detection algorithm has gradually become the mainstream, but in practice, it still needs to save further computing resources and improve the detection efficiency.

In the field of character recognition, character recognition involves two methods, machine learning and deep learning. The traditional machine learning method divides character recognition tasks into feature engineering and classifier. The whole algorithm aims at completing the process of data to information and information. The research is mainly based on decision tree, SVM, neural network and so on.[7] Among them, it's the most important method to use the posterior probabilities of SVM classifier to select samples for classification and recognition. Although it alleviated the relationship between text features and information, it effectively raised the accuracy rate, but the computation process is complex and time-consuming. In addition, machine learning inevitably requires image segmentation. Traditional feature extraction such as edge or orientation is highly complex.

The development of deep learning has replaced the heavy Feature Engineering, and automatically learned the characteristics of images from a large number of tags. CNN is especially eye-catching. In addition to eliminating the flow of manual feature extraction, the way of sharing weights also reduces the number of weights and greatly reduces the computation cost. Although CNN can efficiently extract data local feature information[8], it can not get contextual information. RNN, as a standard model in Natural Language Processing, can handle text context information better. Sutskever et al. [9] mining data sequence information by cyclic recursion property; Liu Pengfei[10] proposes bidirectional

978-1-7281-8288-9/20/$31.00 ©2020 IEEE
DOI 10.1109/AIEA51086.2020.00022

LSTM structure by obtaining longer and bidirectional sequence information. The CRNN model combined with CNN and RNN implements the end to end feature learning method. This method is used for container number detection and provides a new application for intelligent identification technology of container number numbers in the future.

II. PORT CONTAINER NUMBER RECOGNITION SYSTEM

As an application example of OCR technology, port container number recognition system has played a significant role in intelligent port construction. The whole system consists of four parts: sample pretreatment, container number area detection, container number number identification and post-processing recognition correction, as shown in Figure 1. In this paper, the idea of combining the improved yolov3 detection algorithm and the improved crnn recognition algorithm is proposed, which is also in line with the current trend of smart port construction. It can adapt to the current application environment and achieve better results.

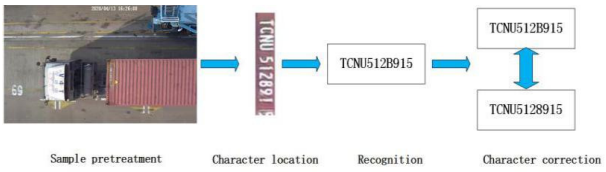


Figure 1. Container recognition process

The whole algorithm flow is shown in Figure 2. In the regional positioning stage, the original image sample to be recognized is preprocessed by data enhancement, then sent to the improved YOLOv3 location network training, and the container number number location model DW-YOLO-13 is obtained. Then the model is located to locate the sample number of the container. After entering the identification stage, the samples obtained from the previous samples are processed by data enhancement, and then sent to the improved CRNN network training. The recognition model Den-CRNN. is finally encapsulated by program and character correction, and an end-to-end container number number recognition system is obtained.

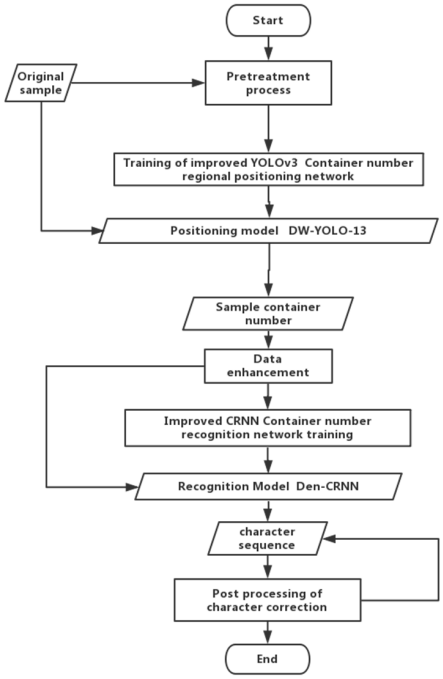


Figure 2. Structure chart of the whole algorithm

III. MODEL STRUCTURE AND OPTIMIZATION

A. Original YOLOv3 model structure

YOLOv3 is developed by the early YOLO algorithm [11]. The main improvements are:

- Adjusting the network structure.
- Object detection using multi-scale features.
- Added residual modules.

Since Prof. He Kaiming proposed the deep residual network [12], the gradient vanishing problem of the deep model has been solved. Therefore, with the idea of RESNET, the depth of the YOLOv3 backbone network is far greater than before. After the convolution + residual modularity combination, the network structure of Darknet -53 is proposed, as shown in Figure 3.

In order to eliminate the negative effect of the pool layer, there is no pool layer and full connection layer in the whole V3 structure. The down sampling of the network is achieved by setting the convolution stride 2. The realization of each volume layer includes convolution, the BN layer and the Leaky relu activation function.BN layer to prevent the network parameter distribution from shifting[13].It has a certain degree of regularization and can effectively avoid the gradient vanishing problem.

	Layers	Filters	Size	Output
	CBL layer	32	3*3	416*416
	CBL layer	64	3*3/2	208*208
1	CBL layer	32	1*1	
	CBL layer	64	3*3	
	Residual			208*208
	CBL layer	128	3*3/2	104*104
2	CBL layer	64	1*1	
	CBL layer	128	3*3	
	Residual			104*104
	CBL layer	256	3*3/2	52*52
8	CBL layer	128	1*1	
	CBL layer	256	3*3	
	Residual			52*52
	CBL layer	512	3*3/2	26*26
8	CBL layer	256	1*1	
	CBL layer	512	3*3	
	Residual			26*26
	CBL layer	1024	3*3/2	13*13
4	CBL layer	512	1*1	
	CBL layer	1024	3*3	
	Residual			13*13

Figure.3 Original YOLOv3 backbone structure

B. improved YOLOv3 structure - DW-YOLO-13

When using YOLOv3 for single class target detection, there are a lot of redundancy in the model. The traditional Darknet53 uses multiple residual units. The ResNet composed of these residual elements is the main reason for the large amount of YOLOv3 network parameters and computation.

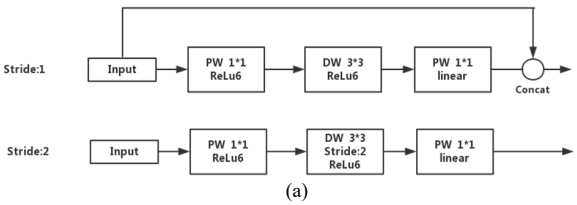
In order to reduce the parameters and speed up, this paper proposes a lightweight network DW-YOLO-13. It based on YOLOv3 to optimize ResNet. Under the condition of ensuring accuracy, three measures are taken:

●Reduce the number of residual units of original YOLOv3 from original 1,2,8,8,4 to 1,2,4,4,2., which greatly reduces the network depth [14].

● Learn from MobileNetV2 to transform some traditional residual units into Inverted Residuals block [15], through the use of depthwise convolution, first dimension and dimension reduction, enhance the propagation of the gradient, and further reduce the amount of computation.

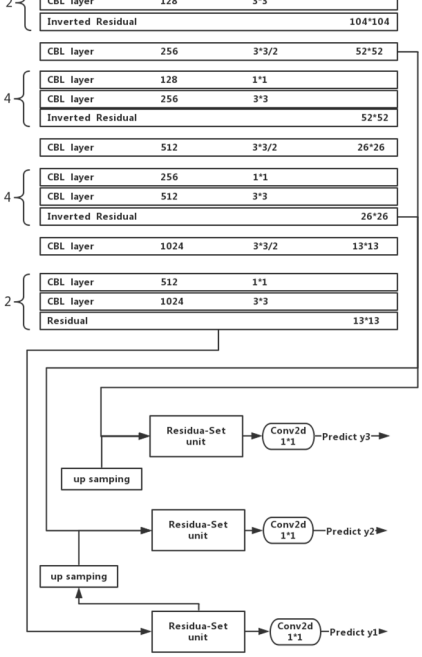
● Reorganize the Convolutional Set unit at the end of the backbone network by joining the 2. The unit Inverted Residuals gets Residua-Set unit to enhance the expression of image features [16].

The overall network structure is illustrated in Figure 4, which is mainly composed of two parts: backbone network Darknet53 and prediction network.

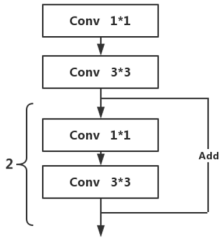


(a)

	Layers	Filters	Size	Output
	CBL layer	32	3*3	416*416
	CBL layer	64	3*3/2	208*208
1	CBL layer	32	1*1	
	CBL layer	64	3*3	
	Residual			208*208
	CBL layer	128	3*3/2	104*104
2	CBL layer	64	1*1	
	CBL layer	128	3*3	
	Inverted Residual			104*104
	CBL layer	256	3*3/2	52*52
4	CBL layer	128	1*1	
	CBL layer	256	3*3	
	Inverted Residual			52*52
	CBL layer	512	3*3/2	26*26
4	CBL layer	256	1*1	
	CBL layer	512	3*3	
	Inverted Residual			26*26
	CBL layer	1024	3*3/2	13*13
2	CBL layer	512	1*1	
	CBL layer	1024	3*3	
	Residual			13*13



(b)



(c)

Figure.4. (a)Inverted Residua block;(b) DW-YOLO-13 model;(c) Residua-Set unit

C. Original CRNN model structure

The CRNN model implements the end to end feature learning mode. The network subject is composed of CNN and RNN [17], which realizes the combination of local feature information and sequence information, and can better describe the data feature information.

The main part of the network is shown in Figure 5. The 7 level CNN is used to extract the local key features. The bidirectional two-level long memory network (LSTM) [18] is used as a variant of the time recurrent neural network to ensure the extraction of the sequence feature information and finally obtain the final sequence through the transcript layer.

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2 × 2, s:1, p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	W × 32 gray-scale image

Figure.5. CRNN backbone model

D. Improved CRNN model— Den-CRNN

Unlike traditional CRNN networks, convolution stacking is also different from ResNet construction. Such as mapping layer operation, the new Den-CRNN network introduced in this paper introduces DenseNet [19] to enhance the information transmission ability of backbone network. As shown in Figure 6, DenseNet is a dense connection network connecting all the layers to merge channels to achieve the reuse of features, which reduces the computation of the network to a certain extent. The goal of reducing redundancy is achieved, so that each layer of the network can get good results by learning only a few features.

In order to reduce the parameters, Den-CRNN does not use the Bottleneck layer and Pooling layer. of the original DenseNet[19], but uses convolution kernel size of 1 * 1 convolution layer to convolution operation of the image, achieving channel fusion and intensive connection.

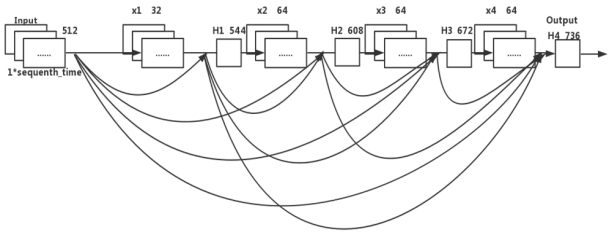


Figure.6. DenseNet model

In the picture, $[X_1, X_2, X_3, X_4]$ Represents the feature map of each level output. $[H_1, H_2, H_3, H_4]$ Representing the Conv+BN+LeakRelu unit of the network layer, the increment of the characteristic map of each layer in DenseNet is set to 32, 64, 64 and 64. respectively.

$$X_l = H_l([X_0, X_1, X_2, ..., X_{l-1}]) \quad (1)$$

Because the semantic information expressed by the network is gradually enriched with the increase of the feature dimension[20]. The greater the meaning of information fusion is, the better the result is. In order to get good results, DenseNet is added to the end of the original network. Finally, the number of the characteristic map numbers is reduced from 736 to 512 by convolution of 1*1.

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	**map: 512 k: 1×1 s:1, p=0**
DneseNet	**map: 736 layers_number=5**
Convolution	#maps:512, k:2 × 2, s:1, p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	W × 32 gray-scale image

Figure.7. Den-CRNN model structure

After the improvement of DW-YOLO-13 and Den-CRNN, the whole container number number recognition algorithm has been formally designed.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The identification model of the container number number is built under the Tensorflow2.0 framework and the DNN module under opencv4.3.0. The Window10 operation system with AMD Ryzen 7 3750H and NVIDIA GeForce GTX1660ti 6GB are used for training.

The sample consists of 4 container number English characters, 6 digit container number numbers, and the last digit check digit. The total number of categories is 11. of the 26000 image samples. In order to ensure that the test data evaluation effect is similar to that of the real scene model, the dataset is divided into training set and validation set. The training dataset is 20000, and the verification set is 1000.5000 sets of test sets, as shown in TABLE I, are trained between the original recognition algorithm and the improved algorithm in this paper. The comparison results between DW-YOLO-13 and 3 original base networks are shown in TABLE II. Compared with the single LSTM network and the original network, the model size contrast is shown in TABLE IV. The comparison experiment uses the exponential attenuation of learning rate, with an initial value of 0.001, compared with the results shown in TABLE II. The attenuation rate is 98%; after continuous tuning, Den-CNN obtains 512 dimensional output sequence; LSTM layer and neuron number are set to 256; training batch period is 50, and each iteration 100 rounds outputs one result.

The recognition effect of the whole container number number is shown in Figure 8.

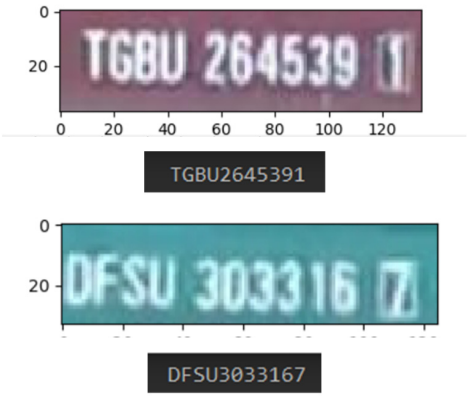


Figure.8. Effect chart of container number number recognition

TABLE I. Data set partitioning information

category	Training set	Validation set	Test set
11	20000	1000	5000

TABLE II.Comparison of dw-YOLO-13 and YOLO models

Method	Backbone network	mAP	FPS
YOLOv2	Darknet19	84.2	13
YOLOv3tiny	____	76.3	32
YOLOv3	Darknet53	98.5	5.8
Dw-YOLO13	DW-Darknet53	97.8	20.8

TABLE III.Comparison of DW-YOLO-13 and YOLO models

Method	BFLOPS	Weight/MB
YOLOv3tiny	8.57	32
YOLOv3	56	234
Dw-YOLO13	12.8	40

TABLE IV.Comparison between Den-CRNN and traditional models

Method	Training time /s	FPS	mAP
LSTM	23800	13	91
LSTM-Attention	27656	10.2	92
CRNN	22582	9	93.6
Den-CRNN	24721	7.5	96.0

From the above TABLE, we can see that at the speed of detection, DW-YOLO-13 is about 25FPS less than YOLOv3tiny of 32fps, while YOLOv3 is 5.8fps, indicating that the YOLOv3 speed of lightweight MobileNetV2 network is 4-5 times faster. On the model size, the size of DW-YOLO-13 model is 40MB, which is nearly 6 times compression than that of YOLOv3.So that DW-YOLO-13 can run on the embedded platform that YOLOv3 can not deploy.

In terms of recognition, Den-CRNN has increased the accuracy rate by 2-3 percentage points at the expense of some speed, which proves the superiority of the model. It can be seen that, based on the improved YOLO+CRNN network model, the container number area detection achieves a 97.8% accuracy rate while maintaining the same marking mode, and the overall recognition rate is 96%.

V. CONCLUSION

In this paper, a container number recognition algorithm combining improved lightweight YOLOV3 and improved CRNN is proposed, which has faster network learning ability and generalization ability of models. Although the recognition effect of the model is improved, the design of CRNN network structure is complex. At the same time, the introduction of DensNet requires a certain amount of computation. Therefore, it has not achieved good results in real-time. Therefore, the next problem to be further studied is: a) to continue lightweight design for Den-CRNN networks and improve real-time performance; b) properly introduce attention mechanism to selectively focus on key output sequence information instead of every moment.

REFERENCES

[1] Li H, Wang P, Shen C. Towards End-to-end Text Spotting with Convolutional Recurrent Neural Networks[J]. 2017.

[2] Zhuo junfei, hu yu. Research on license plate location algorithm based on edge detection and projection method [J]. Chinese science and technology bulletin,2010,26(03):438-441.

[3] Wang yan, xie guangsu, shen xiaoyu. Research on a new license plate detection and recognition method based on MSER and SW [J]. Acta metrology sinica,2019,40(01):82-90.

[4] Ren S, He K, Girshick R B, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2017, 39(6): 1137-1149.

[5] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6517-6525.

[6] RedmonJ, Farhadi A. YOLOv3: An Incremental Improvement[C]//IEEE Conference on Computer Vision and Pattern Recognition, arXiv preprint arXiv: 1804.02767, 2018.

[7] Rajvanshi N, Chowdhary K R, Rajvanshi N, et al. Comparison of SVM and naive Bayes text classification algorithms using WEKA [J].International Journal of Engineering & Technical Research, 2017, 6 (9):141-143.

[8] Kim Y. Convolutional neural networks for sentence classification [J]. arXiv preprint arXiv, 2014: 1408. 5882.

[9] Sutskever I, Martens J, Hinton G E. Generating text with recurrent neural networks [C]// Proc of International Conference on Machine Learning. 2011: 1017-1024.

[10] Liu Pengfei, Qiu Xipeng, Huang Xuanjing. Recurrent neural network for text classification with multi-task learning [J]. arXiv preprint arXiv, 2016: 1605. 05101.

[11] Guo Jinxiang, Liu Libo,Xu Feng, et al. Airport Scene Aircraft Detection Method Based on YOLO v3[J]. Laser & Optoelectronics Progress, 2019, 56(19): 191003

[12] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//IEEEConference Computer Vision and Pattern Recognition, 2016: 770-778.

[13] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]//The 32nd International Conference on Machine Learning.[S.l.:s.n.], 2015:448-456.

[14] Cui Jiahua,Zhang Yunzhou,Wang Zheng, et al. Light-Weight Object Detection Networks for Embedded Platform[J]. Acta Optica Sinica, 2019, 39(4): 0415006

[15] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmenta- tion. arXiv preprint arXiv:1801.04381 (2018)

[16] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[17] Shi Baoguang, Xiang Bai, Cong Yao. An end-to-end trainable neuralnetwork for image-based sequence recognition and its application to scene text recognition [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2015, 39 (11): 2298-2304.

[18] Hochreiter S, Schmidhuber J. Long short-term memory [J]. NeuralComputation. 1997, 9 (8): 1735-1780.

[19] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks-[C]//Proceedings of the IEEE conference on computer vision and pattern recognition, 2017:4700-4708.

[20] Zhang R, Li W, Mo T. Review of Deep Learning[J]. Information and Control, 2018,47(4):385-397.