

# A Method of License Plate Location and Character Recognition based on CNN

Wei Fang<sup>1,2,3\*</sup>, Weinan Yi<sup>1</sup>, Lin Pang<sup>1</sup>, and Shuonan Hou<sup>4</sup>

<sup>1</sup> School of Computer & Software, Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science & Technology, China

<sup>2</sup> State Key Lab. for Novel Software Technology, Nanjing University, China

<sup>3</sup> Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, China

<sup>4</sup> Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin, People's Republic of China

\*Corresponding author: Wei Fang

E-mail: Fangwei@nuist.edu.cn

*Received March 17, 2020; revised June 21, 2020; revised July 20, 2020; accepted July 21, 2020;  
published August 31, 2020*

---

## Abstract

At the present time, the economy continues to flourish, and private cars have become the means of choice for most people. Therefore, the license plate recognition technology has become an indispensable part of intelligent transportation, with research and application value. In recent years, the convolution neural network for image classification is an application of deep learning on image processing. This paper proposes a strategy to improve the YOLO model by studying the deep learning convolutional neural network (CNN) and related target detection methods, and combines the OpenCV and TensorFlow frameworks to achieve efficient recognition of license plate characters. The experimental results show that target detection method based on YOLO is beneficial to shorten the training process and achieve a good level of accuracy.

---

**Keywords:** CNN, YOLO, character recognition, license plate recognition, target detection

---

This work was supported in part by the Open Research Project of State Key Laboratory of Novel Software Technology under Grant KFKT2018B23, the Priority Academic Program Development of Jiangsu Higher Education Institutions, the 2018 Tiancheng Huizhi Innovation Promotion Education and Scientific Research Innovation Fund of the Ministry of Education under Grant 2018a03038 and the Open Project of Provincial Key laboratory for Computer Information Processing Technology under Grant KJS1935, Soochow University.

## 1. Introduction

With the continuous improvement of the economic level and the development of science and technology, the number of small passenger cars and private cars in China has increased from 226 million in 2014 to 390 million in 2018, an increase of 72%. Obviously, for vehicles, management work has also become more and more arduous. The license plate can be effectively used as information for managing the vehicle. If it is solely dependent on the human eye, the efficiency is low and the correct rate is not high, so the generation of license plate technology is necessary. License plate recognition technology is also an important part of modern intelligent transportation systems [1], and its application is very extensive. Based on computer vision processing, digital image processing, pattern recognition and other technologies, it processes and analyzes vehicle images or video images captured by cameras or cameras to obtain the license plate number of each vehicle, thereby completing the identification process. License plate recognition has the potential to create social and economic benefits, not only can bring a wide range of convenience to people's lives, but also has a universal application in vehicle management. Since the image recognition technology has become more and more mature, the amount of data used for training has been continuously expanded, and the image processing capability has been continuously improved, the license plate recognition technology has also been developed and matured. It can free people from tedious and repeated observations and tests, and can Greatly improve accuracy. The license plate recognition technology has a wide range of applications in traffic flow monitoring, vehicle entry and exit time, checkpoint vehicle monitoring, and fixed-point vehicle and public vehicle anti-theft. [2]. The implementation process of the traditional license plate recognition algorithm [3] is shown in Fig. 1.

With the popularization of artificial intelligence, artificial intelligence deep learning technology has been widely used in license plate recognition, such as convolutional neural network detection and recognition, variable length license plate end-to-end recognition model, sample generated GAN confrontation network. Based on the deep learning license plate recognition method, the model is simple, but the high-performance hardware platform provides accurate calculation support and optimized network, but the accuracy is not low [4]. Based on the end-to-end license plate recognition method of convolutional neural network, with the support of a large amount of data, the deep neural network is used to extract the license plate character feature to obtain the training model. The detection of license plates does not require character segmentation and is robust to the angle of illumination. Using deep learning for license plate recognition, the framework is relatively simple, and in the case of strong hardware performance and sufficient training samples, it can obtain better recognition effect than traditional algorithms in a short time.

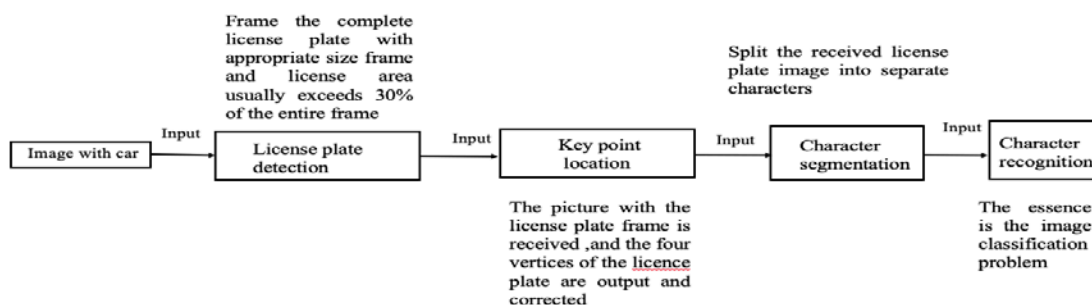


Fig. 1. Traditional license plate recognition algorithm implementation process

This paper mainly does the following work:

- (1) Inspired by the single-stage detector, we used the deep learning-based target detection method, by modifying the configuration parameters of YOLO, preprocessing the training set and the test set, the position of the license plate is located in a picture containing the license plate.
- (2) In the license plate character recognition section, we used the convolutional neural network and the TensorFlow framework, the license plate characters are trained to recognize the license plate characters.
- (3) Experiment compared the speed of license plate localization. The method we proposed to use YOLO detect license plate method, compared with the R-CNN minus R[5], Fast R-CNN[6] and Faster R-CNN[7] positioning method of license plate localization speed, respectively. The experimental results show that our method is faster in speed than other three methods, average of nearly six times, which compared with the Fast R-CNN is nearly 70 times faster.

## 2. Related Work

### 2.1 Deep Learning

When it comes to deep learning, we should start with the idea of artificial intelligence. Artificial intelligence, also known as machine intelligence, was presented at the Dartmouth conference in 1956. As a cutting-edge science and research that explores and simulates the wisdom of human beings, artificial intelligence has undergone a difficult and tortuous development in nearly half a century. With the development of artificial intelligence technology, the use of artificial intelligence can be seen everywhere in the world. Typical areas include symbolic and logical reasoning, machine learning, pattern recognition, and automation engineering control. Deep learning is a powerful technology for developing artificial intelligence. In the era of big data, more complex models can fully exploit the valuable information contained in massive data. The essence of deep learning refers to a series of measures to effectively train neural networks with deep structures, the source of which is the knowledge of artificial neural networks [8]. The MP model, established by psychologist McCulloch and mathematical logician Pitts in 1943, was the earliest neural network, and although it was impossible to complete the learning task, it was the beginning of learning artificial neural networks. In the following 50 years, many researches on artificial neural networks have been made. In 1949, Hebb first proposed the idea of learning about biological neural networks. In 1958, Rosenblatt proposed the perceptron model and its learning algorithm. In the 1980s and 1990s, it reached the world climax of the discussion of neural networks. The Boltzmann machine, the Hopfield neural network and the multilayer perceptron are the most interesting models of the period.

It is widely believed that deep learning began in 2016 with the beginning of two papers published by Hinton et al. [9,10]. Since then, a large number of deep learning models have received extensive attention, including Deep Belief Net, Sum-Product Network, Convolutional Neural Network (CNN), and generated confrontation. Generative Adversarial Network (GAN) [11] and so on.

Combine mature deep learning models with appropriate training techniques such as max pooling, dropout, and drop connect. Artificial intelligence is already in handwritten digit

recognition, ImageNet classification, and speech recognition. Even in the field of recognition of ancient words such as Telugu [12] and many other historical achievements.

## 2.2 Convolutional Neural Networks

In recent years, Convolutional Neural Networks (CNN) has been widely used in image understanding such as face recognition [13], image classification [14], sentence classification [15]. The history of deep learning has played a pivotal role in the development of computer vision research and the performance of the century has played a role that cannot be replaced by other methods, especially with the development of large-scale image data and the rapid development of computer hardware (especially GPU). Convolutional neural networks have achieved significant results in image understanding [16, 17]. Convolutional neural networks receive models inspired by the neural mechanisms of the visual system. The core idea is local receptive fields and weight sharing. The convolutional neural network can also be combined with the DCGAN network to increase

the training samples to achieve the purpose of increasing the recognition accuracy. This can be used for difficult collection of extremely severe weather samples [17], gesture recognition [18] and so on.

A general convolutional neural network has four layers: an input layer, a convolution layer, a sampling layer, and an output layer. The input is a two-dimensional image. The convolutional layer and the sampling layer alternately appear multiple times throughout the network. It is the most important level. The output layer is the full connection mode of the feedforward network, and the dimension is the number of categories in the classification task.

## 2.3 Object detection depend on deep learning

In the target detection method, the deep learning method [19] and the traditional method have significant differences in representation and model. From the point of view of representation, the traditional method mainly adopts the low-level visual features of manual design. The characteristics of this manual design rely on empirical knowledge on the one hand, and mainly design for specific categories. At the same time, the manual design method has weak expression ability and it is difficult to describe the complex changes of objects. The deep learning method adopts the data-driven representation learning mechanism, and adaptively constructs the feature extractor according to the training data, which often requires a large amount of labeled data, and it is difficult to explain in the learning process and the learned representation. From the model point of view, traditional methods often use linear or simple nonlinear models. These models tend to be weak in modeling. For complex object changes, multiple model fusions are often required to achieve higher precision. Moreover, these models are difficult to extend to multi-class target detection, and it is necessary to separately learn the classifier for each category. In contrast, the deep learning method generally adopts a highly nonlinear model, so it can effectively model the complex change pattern of the object, and can also be relatively simple to expand from a single category to multiple categories. However, since this model is highly nonlinear and complex, the computational complexity of the model is high, resulting in a slower overall detector speed.

Target detection based on deep learning can be divided into two typical types of detection methods [20], one is a two-stage detector, which can be typically divided into two stages during the detection process. The first stage can generate possible candidate regions containing objects, and the second stage is to further classify and calibrate the candidate regions to obtain the final test results. The second type of detection method is a single-stage detector. In contrast, such detectors directly give the final detection result without

simultaneously displaying the candidate region. In the design of this paper, YOLO is used in the single-stage detector. It uses the network format detection method to synthesize the information of each position by combining the information of the whole picture.

## 2.4 YOLO

YOLO (You Only Look Once) [21], as the name suggests, YOLO can do the test results for a given picture with only one look. The specific method of YOLO detection is to divide the image into grids and then predict the border of the object in each grid. YOLO first scales the input image to the same size, and then divides the image into a grid. Then the detection frame of each object will fall in one of the grids, then YOLO will input the image to a CN and pass it. Predicting each detection box will result in a corresponding probability and x, y, w, h of the detection box.

YOLO takes the whole image as input. It is no different from the ordinary convolutional neural network except the output part. The output layer is directly connected to the feature map obtained by the previous layer through the full connection, and then the prediction is needed. Tensor. Since YOLO uses a fully connected layer, its prediction of the position of each object is based on the characteristics of the whole picture as input, which will bring the advantage of sufficient context-assisted judgment.

Up till now, the YOLO family consists of three members, namely, YOLO, YOLO v2, YOLO v3. Research on YOLO has been written above, followed by a brief explanations of the YOLO v2 and v3's improvements.

YOLO v2 [22] uses anchor on the basis of Yolo, but it is not selected manually. Instead, it is learned through clustering. K-means algorithm is used on the training set, aiming to make the loss of large box and small box equal. YOLO v2 uses a new basic network structure based on Googlenet named Darknet-19, which contains 19 convolution layers and 5 max pooling layers, and uses batchnormalization to accelerate convergence. At the same time, YOLO v2 carries out a joint training during the training process. Additionally, data sets containing only tag information are used for classification training, so as to expand the number of objects that can be predicted by the network.

One improvement of YOLO v3 [23] is the multi-scale budget. Boundary box prediction still uses dimension clusters as anchor boxes in YOLO v2 to predict the boundary boxes. During training, YOLO v3 uses the sum of squared error losses. YOLO v3 uses logistic regression to predict the object score of each bounding Box. If the previous bounding box overlaps the Ground Truth object more than any other bounding box before it, this value should be 1. If the previous bounding box is not the best, but does overlap the ground truth object above a certain threshold, YOLO v3 will ignore this prediction. YOLO v3 uses a threshold of 0.5. Unlike [15], YOLO v3 systems only assign one bounding box for each ground Truth object. If previous bounding box is not assigned to grounding box objects, the coordinates or category forecast will not cause damage.

## 3. Proposed Method

### 3.1 Selection of data

The training data set and the verification data set constitute the data set required by YOLO, and the training data is used to train the model while the obtained data is appropriately adjusted by the verification data, both of which need to contain pictures and labels. First, you need to label each image in the two data sets to get the corresponding xml file. Since the Xml

file is not useful, you need to convert it to the txt label text that saves the absolute path of the image. Therefore, we choose to use the annotation tool. LableImg made a label file for the image. This has an XML file marked in VOC format. Next, you need to convert the xml file to a txt file. I chose to use batch processing for conversion.

In realizing the recognition of license plate characters, we know that Chinese license plates are composed of provincial Chinese characters, Arabic numerals and English. The Arabic numerals are 10 Arabic numerals of 0-9, and there are 24 English words (excluding O and I). Beijing, Guangdong, Jiangsu, Shanghai, Nanjing, Hong Kong, Macao, Taiwan and so on. The license plate consists of seven characters. The first character is Chinese characters, the second one is English letters, and the last five digits are randomly combined with Arabic numerals and English.

It is impossible to achieve absolute sample balance in this task. It is possible to achieve a general equilibrium by artificially controlling the difference in the number of samples. This paper only realizes the character recognition of the six provinces of Beijing, Fujian, Jiangsu, Guangdong, Zhejiang and Shanghai.

### 3.2 Modify the YOLO configuration file

First, create a .names file in the YOLO home directory data folder to hold the detected category name and save a category in one row. Because we are mainly used to detect license plates, write IDs on the first line. Then, modify the voc.data file in the cfg directory. As shown in [Table 1](#), the number of detection categories, the absolute path of the training set, the absolute path of the verification set, the name of the detection category, and the name used to save the training model are modified from bottom to bottom. address. Then change the classes in [region] in the cfg/yolo-voc.cfg file to 1 because there is only one license plate category; the filters of the convolutional layer adjacent to [region] are given by:

$$\text{Filters} = \text{Num} * (\text{classes} + \text{coord} + 1) \quad (1)$$

**Table 1.** Modified voc. data

classes	1
train	trainImagePath.txt
valid	validImagePath.txt
names	plate.names
backup	backup

### 3.3 YOLO's loss function

YOLO's loss function consists of loss of the central coordinate of the prediction, loss of the width and height of the prediction bounding box, loss of the predicted category and loss of the confidence of the prediction. The first is to make a loss on the predicted center coordinates. Its formula is as follows:

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B l_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \quad (2)$$

When the predicted object is included in the grid  $i$ , the predicted value of the  $j$ th prediction frame is valid for the prediction, and the value of  $l_{ij}^{obj}$  is set to 1, otherwise if the prediction target is not included in the grid  $i$ , then  $l_{ij}^{obj}$  is 0. In principle, each prediction bounding box is only responsible for predicting a target, so depending on which prediction box has the highest real-time IOU to determine the target it is responsible for.  $(x, y)$  represents the position of the predicted bounding box, and  $(\hat{x}, \hat{y})$  represents the actual position obtained from the training. The next step is to make a loss on the width and height of the predicted bounding box. Its



formula is as follows:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \quad (3)$$

Then there is a loss to the predicted category, and its formula is defined as follows:

$$\sum_{i=0}^{S^2} l_i^{obj} \sum_{j=0}^B [(p_i(c) - \hat{p}_i(c))^2] \quad (4)$$

This is similar to the summed squared error of the classification. Finally, the loss of confidence in the prediction is made. Its formula is defined as follows:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{obj} [(C_i - \hat{C}_i)^2] + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{noobj} [(C_i - \hat{C}_i)^2] \quad (5)$$

$C$  is the score of confidence, and  $\hat{C}$  is the intersection of the predicted bounding box and the basic fact. As with the definition of  $l_{ij}^{obj}$  in Equation 2, the value is 1 when there is a target in the cell, if there is no target, the value of  $l_{ij}^{obj}$  will be set 0. The parameter  $\lambda$  plays a key role in improving the stability of the model, and is used for different weighting values of the loss function.

### 3.4 License plate character data set

China's license plate has the province's Chinese characters, Arabic numerals and English. The Arabic numerals are 10 Arabic numerals from 0-9, and there are 24 English words (excluding O and I). The province's Chinese characters are Beijing, Guangdong, Jiangsu, Shanghai, Nanjing, Hong Kong, Macao, Taiwan, etc. The license plate consists of seven characters, the first character is Chinese characters, the second one is English letters, and the last five digits are randomly combined with Arabic numerals and English.

It is impossible to achieve absolute sample balance in this task. It is possible to achieve a general equilibrium by artificially controlling the difference in the number of samples. This paper only realizes the identification of six identity characters in Beijing, Fujian, Jiangsu, Guangdong, Zhejiang and Shanghai. The prepared data set is shown in [Fig. 2](#) and [Fig. 3](#). There are a total of 34 categories, each with about 170 sample images, and the data set comes from the network.

### 3.5 Convolutional neural network parameter initialization and training

The convolutional neural network training process, the parameter learning process, consists of two phases: forward propagation and back propagation. Forward propagation reflects the transfer of feature information, and backpropagation reflects the correction of the learned model and the error information. The network weights need to be initialized before training, such as the size of the input picture (usually compressed to 2<sup>nth</sup> power size), the parameters of the convolution kernel, and so on. Then the input data passes through the convolution layer, and the pooled layer and the fully connected layer get the output value, and the error is compared with the target value. If the error is greater than expected, it will be expected to transmit back to the network in reverse, thereby obtaining the error of each layer network, and then updating the weight. The above process is repeated until the appropriate weight is obtained such that the error is within the desired range.



Fig. 2. A test set containing English letters and Arabic numerals



Fig. 3. Training set containing province abbreviation

There are three convolutional neural network models used in this paper to identify the license plate letters, license plate Arabic numerals and provincial Chinese abbreviations. The essence is the classification problem. Because the number of categories is different, there are only differences in the output layers between the models. Explain the model for identifying the abbreviation of Chinese characters in the license plate province. The convolutional neural network model I designed contains two layers of convolution. The convolution method is SAME, and then a fully connected layer is added. The output is 6 types (Shanghai, Beijing, Fujian, Jiangsu, Guangdong, Zhejiang). The model structure is shown in Fig. 4, we propose a Chinese characters training algorithm, as shown in Algorithm 1.

Algorithm 1 Chinese characters training algorithm

```

num_classes = 6
batch_size=60
iterations=300
for i = 0, i < num_classes do
    width = img.size[0]
    height = img.size[1]
    for h = 0, h < height do
        for w = 0, w < width do
            if pixel value > 230 then
                w+h*height = 1
            else w+h*height = 0
            end if
        end for
    end for
end for
define Convolution layer1
define Convolution layer2
define Fully connected layer
//make the line thin to improve accuracy

```

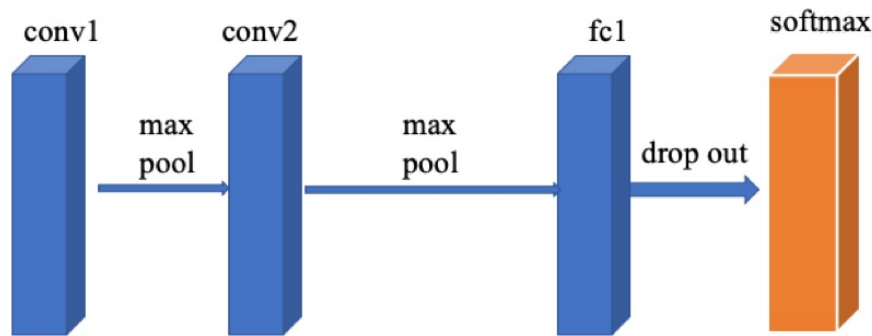


```

define optimizer
for it = 0, it < 300 do
  if accuracy >= 0.9999 and it >= 150 then
    break
  end if
end for

```

The first layer of convolutional layer receives the original image, the image size is 32x40x1, it has 16 8x8 convolution kernels, and the step size is 1. There are 32 5x5 convolution kernels in the second layer of convolutional layer, with a step size of 1. The filter size of the first layer of the pooling layer is 2x2, the step size is 2, the filter size of the second layer is 1x1, the step size is 1, and all pooling methods are SAME. Behind these connections, a layer of fully connected layers is set up to set the image data dimension to 1, which has 512 kernels. The final output divides the picture into six possible outcomes.



**Fig. 4.** Classification model structure diagram

## 4. Analysis of experimental results

### 4.1 License plate location

The environment we use is Ubuntu-18.04.2, OpenCV3.2.0, YOLO. The type and number of pictures are shown in **Table 2**. After making the labels for all the training pictures and verification pictures, save them in xml format. The process of labeling is to use the LabelImg visual tool to frame the license plate area in the picture and give it the type. In this paper, the type is plate. Then through the batch operation, the xml file is converted into a txt file, so that YOLO addressing training.

We can train directly, or we can download the pre-training model provided by YOLO to speed up the training. We speed up the training model by using the darknet19\_448.conv.23 pre-training model provided by YOLO. The final rendering is shown in **Fig. 5**. It can be seen that the license plate is more framed.

**Table 2.** Image sets were used in paper

Image set	Number of Image
Train	4k
Test	1k

**Fig. 5.** Model test

#### 4.2 License plate character recognition

The convolutional neural network model used to identify characters is shown in Fig. 4. In the training, because the number of training pictures is different, the number of pictures in each batch is different. The model used to train Arabic numerals and provincial Chinese abbreviations is 1000 times, and 60 pictures are trained per batch. The model used to train the letters, the number of iterations is 500, and 100 pictures are trained per batch.

The province's abbreviated character recognition results are shown in Table 3.

**Table 3.** Province abbreviation character recognition result

Character	苏	沪	浙
苏	0.98	0.01	0
沪	0.02	0.99	0.02
浙	0	0	0.98

The city code identification result is shown in Table 4.

**Table 4.** City code recognition result

Character	B	E	G
B	0.98	0.02	0.02
E	0.01	0.97	0
G	0.01	0.01	0.98

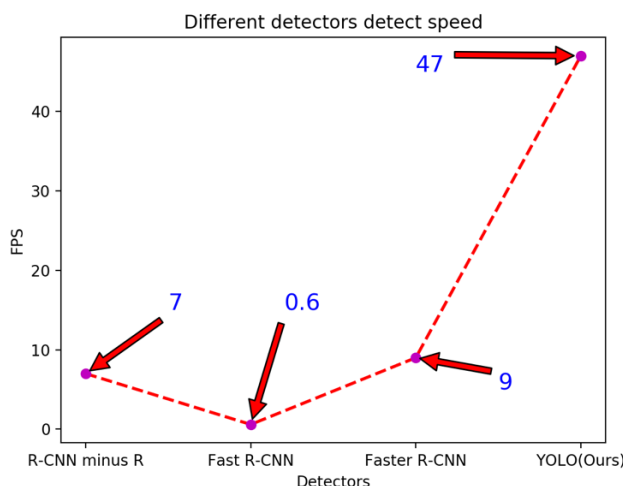
#### 4.3 Detection speed comparison

We compared the license plate location model based on YOLO with the model based on R-CNN minus R, Fast R-CNN and Faster R-CNN in terms of recognition speed. The experimental results are shown in Fig. 6.

Compared with other two-stage detectors, the advantages of single-stage detector in detection speed can be found. Compared with R-CNN minus R, the YOLO localization method proposed by us is nearly 7 times faster, 80 times faster than Fast R-CNN and 7 times

faster than Faster R-CNN, so it can be seen that our proposed method is efficient. There are mainly the following reasons for such a big promotion:

- 1、One single-stage detector, such as YOLO, directly gives the final detection result based on the whole input image;
- 2、Compared to two-stages detector, single-stage detector does not show the steps to generate candidate regions.



**Fig. 6.** Detect speed

## 5. Conclusions

This paper proposes a method to modify the YOLO configuration parameters and pre-process the near 5k license plate dataset. Experimental result shows that our method achieved a higher speed on the license plate location of the license plate image than other methods, it shows that the single-stage detector is useful in the application of license plate location. And then proposes a convolutional neural network using three-layer convolution. The method of accumulating neural networks also achieves a higher accuracy rate in the recognition of license plate characters. In future research, consideration will be given to improving the generalization ability of the model, at the same time, it makes up for the problem that the single-stage detector YOLO cannot combine speed and accuracy, thereby improving the accuracy of the recognition of complex shape license plates.

## References

- [1] Anagnostopoulos C N E, Anagnostopoulos I E, Loumos V, et al, "A license plate-recognition algorithm for intelligent transportation system application," *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp.377-392, 2006. [Article \(CrossRef Link\)](#)
- [2] Duan T D, Du T L H, Phuoc T V, et al, "Building an automatic vehicle license plate recognition system," in *Proc. of Int. Conf. Comput. Sci. RIVF*, pp.59-63, 2005. [Article \(CrossRef Link\)](#)
- [3] Parisi R, Di Claudio E D, Lucarelli G, et al., "Car plate recognition by neural networks and image processing," in *Proc. of ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187)*, vol. 3, pp.195-198, 1998. [Article \(CrossRef Link\)](#)

- [4] Zang D, Chai Z, Zhang J, et al, "Vehicle license plate recognition using visual attention model and deep learning," *Journal of Electronic Imaging*, vol. 24, no. 3, pp.33001-33001, 2006. [Article \(CrossRef Link\)](#)
- [5] Lenc K, Vedaldi A, "R-CNN minus R," in *Proc. of British Machine Vision Conference*, vol. 5, no. 12, pp.1-12, 2015. [Article \(CrossRef Link\)](#)
- [6] Girshick R, "Fast R-CNN," in *Proc. of the IEEE international conference on computer vision*, pp.1440-1448, 2015. [Article \(CrossRef Link\)](#)
- [7] Ren S, He K, Girshick R, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp.1137-1149, 2017. [Article \(CrossRef Link\)](#)
- [8] LeCun Y, Bengio Y, Hinton G, "Deep learning," *Nature*, vol. 521, no. 7553, pp.436-444, 2015. [Article \(CrossRef Link\)](#)
- [9] Hinton G E, Osindero S, Teh Y W, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp.1527-1554, 2006. [Article \(CrossRef Link\)](#)
- [10] Hinton G E, Salakhutdinov R R, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp.504-507, 2006. [Article \(CrossRef Link\)](#)
- [11] Goodfellow I, Pouget-Abadie J, Mirza M, et al, "Generative adversarial nets," *Advances in neural information processing systems*, pp.2672-2680, 2014. [Article \(CrossRef Link\)](#)
- [12] Achanta R, Hastie T, "Telugu OCR framework using deep learning," *arXiv preprint arXiv:1509.05962*, 2015. [Article \(CrossRef Link\)](#)
- [13] Lawrence S, Giles C L, Tsoi A C, et al, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp.98-113, 1997. [Article \(CrossRef Link\)](#)
- [14] Krizhevsky A, Sutskever I, Hinton G E, "Imagenet classification with deep convolutional neural networks," *Communications of The ACM*, vol. 60, no. 6, pp.84-90, 2017. [Article \(CrossRef Link\)](#)
- [15] Kim Y, "Convolutional neural networks for sentence classification," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1746-1751, 2014. [Article \(CrossRef Link\)](#)
- [16] Wei Fang, Yewen Ding, Feihong Zhang, Victor S. Sheng, "DOG: A New Background Segmentation Recognition Method based on CNN," *Neurocomputing 361 (2019)*, pp.85-91, 2019. [Article \(CrossRef Link\)](#)
- [17] Wei Fang, Feihong Zhang, Victor S. Sheng, Yewen Ding, "A Method for Improving CNN-Based Image Recognition Using DCGAN," *CMC: Computers, Materials & Continua*, vol. 57, no. 1, pp.167-178, 2018. [Article \(CrossRef Link\)](#)
- [18] Wei Fang, Yewen Ding, Feihong Zhang, Victor S. Sheng, "Gesture recognition based on convolutional neural network for calculation and text output," *IEEE access*, vol. 7, pp.28230-28237, 2019. [Article \(CrossRef Link\)](#)
- [19] Erhan D, Szegedy C, Toshev A, et al, "Scalable object detection using deep neural networks," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.2147-2154, 2014. [Article \(CrossRef Link\)](#)
- [20] Lin T Y, Goyal P, Girshick R, et al, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp.318-327, 2020. [Article \(CrossRef Link\)](#)
- [21] Redmon J, Divvala S, Girshick R, et al, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.779-788, 2016. [Article \(CrossRef Link\)](#)
- [22] Redmon, Joseph, and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.7263-7271, 2017. [Article \(CrossRef Link\)](#)
- [23] Redmon, Joseph, and Ali Farhadi. "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018. [Article \(CrossRef Link\)](#)



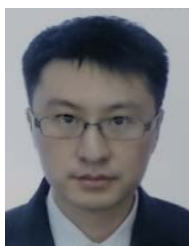
**Wei Fang** is currently an Associate Professor in the Jiangsu Engineering Center of Network Monitoring at the NanJing University of Information Science & Technology in China, also with State Key Laboratory for Novel Software Technology, Nanjing University. He received his Ph.D and M.Sc degrees from Soochow University all in Computer Science. His research interests are in the areas of Cloud Computing, Big Data, Deep Learning, and Cloud Computing. He is a PC member for a number of international conferences and a reviewer for several international journals.



**Weinan Yi** received the B.S. degree in software engineering from the Nanjing University Jinling College, China, in 2019. He is now pursuing a M.S. degree in software engineering at Nanjing University of Information Science & Technology, Nanjing, JiangSu, China. His research interests include the deep learning and image recognition.



**Lin Pang** received the B.S. degree in computer science and technology from Jining Medical University, Jining, ShanDong, China, in 2019. She is now pursuing a M.S. degree in software engineering at Nanjing University of Information Science & Technology, Nanjing, JiangSu, China. Her research interests include deep learning, reinforcement learning and image processing.



**Shuonan Hou** is a Ph.D student. He was born in 1984. He has received the M.S degree in power systems and automation from Northeastern University, Liaoning, China. He is currently pursuing the Ph.D. degree in the School of Electrical Engineering, Tianjin University, Tianjin, China. His research interests include reliability and risk assessments of power system, integrated energy system, and smart grid.