

FAFNet: A False Alarm Filter Algorithm for License Plate Detection Based on Deep Neural Network



Hui Huang¹, Zhe Li^{2*}

¹ Guangxi Science & Technology Normal University, Laibin 546199, China

² Institute of Economics and Management, Hubei Engineering University, Xiaogan 432000, China

Corresponding Author Email: lizhe_lz@hbeu.edu.cn

<https://doi.org/10.18280/ts.380525>

ABSTRACT

Received: 5 May 2021

Accepted: 19 August 2021

Keywords:

YOLOv5, FAFNet, false alarm filter, model generalization, embedded device

The license plate detection technology has been widely applied in our daily life, but it encounters many challenges when performing license plate detection tasks in special scenarios. In this paper, a license plate detection algorithm is proposed for the problem of license plate detection, and an efficient false alarm filter algorithm, namely the FAFNet (False-Alarm Filter Network) is proposed for solving the problem of false alarms in license plate location scenarios in China. At first, this paper adopted the YOLOv5 target detection algorithm to detect license plates, and used the FAFNet to re-identify the images to avoid false detection. FAFNet is a lightweight convolutional neural network (CNN) that can solve the false alarm problem of real-time license plate recognition on embedded devices, and its performance is good. Next, this paper proposed a model generalization method for the purpose of making the proposed FAFNet be applicable to the license plate false alarm scenarios in other countries without the need to re-train the model. Then, this paper built a large-scale false alarm filter dataset, all samples in the dataset came from the industries and contained a variety of complex real-life scenarios. At last, experiments were conducted and the results showed that, the proposed FAFNet can achieve high-accuracy false alarm filtering and can run in real-time on embedded devices.

1. INTRODUCTION

As China's car ownership has increased dramatically in recent years, license plate recognition is playing an increasingly important role in our lives. Now, it is widely used in all aspects such as traffic management, digital security monitoring, vehicle identification, and urban parking management, etc. During license plate recognition, the first task is to locate and detect the license plate, and the recognition is the second job. Therefore, the results of license plate location [1-4] will directly affect the subsequent recognition works, so it is an important part in the task of license plate recognition.

Existing target detection algorithms are mainly divided into two categories: traditional detection algorithms and the detection algorithms based on deep learning. The traditional detection algorithms generally use a sliding window to search for objects of interest in images or videos, extract the features of the areas of interest, and then use the traditional machine learning algorithms to classify the features, thereby completing the target detection jobs. However, traditional target detection algorithms only have low accuracy and low robustness for detection tasks in complex scenarios.

The detection algorithms based on deep learning is developing in two main directions: two-stage algorithms such as the R-CNN series, and one-stage algorithms such as YOLO [5], and SSD [6]. The main difference between the two is that the two-stage algorithms need to generate a proposal (a preselection box that may contain the objects to be inspected) at first, and then perform fine-granularity object detection;

while the single-stage algorithms directly extract features from the network to predict the class and location of the objects. Compared with the traditional target detection algorithms, detection algorithms based on deep learning can adapt well to different detection scenarios, and can perform target detection in the environment of GPUs and other devices in real time, and among the deep learning detection algorithms, YOLOv5 is the one with the highest detection efficiency and accuracy.

Although the existing deep learning-based target detection algorithms are quite mature already, still they are facing many challenges in license plate location tasks, and one of the challenges is the false alarm problem. A false alarm is that, during license plate recognition, an area that is not a license plate has been recognized as a license plate, and this can bring great inconvenience to urban parking management. The main reason of false alarms is the complex background texture of the non-license plate area and the similarity between the characters in various scenes and the features of the license plate area. Figure 1 shows some false alarm scenarios. The false-alarmed "license plates" are marked by yellow boxes.

Aiming at the above-mentioned false alarm problem, this paper proposed the FAFNet algorithm. Although the real license plate area and the false license plate area have similar features, the area around real license plates and the area around false license plates have different features: the area around real license plates is mainly characterized by the front of the car, while the area around false license plates has other non-car front features; then, by utilizing this difference in features, we can realize the classification of real and false license plates. This paper used FAFNet to classify real and false license plate

images and obtained satisfactory results. In our research, a reasonable area was captured according to the license plate coordinates and related capture parameters obtained from license plate recognition, then, the image of this area was input into the FAFNet for false alarm judgment. Figure 2 gives a diagram of the false alarm filter process of license plate recognition.

The realization of FAFNet is based on deep CNN, and it could perform end-to-end training. Research in recent years shows that, although CNN has made great progress in computer vision tasks such as image classification and object detection, still, it faces the challenge of running on embedded devices. FAFNet is a very efficient lightweight neural network, it only needs 0.024GFLOPs to complete a forward propagation operation, and it can be deployed on embedded devices to achieve real-time false alarm filter.

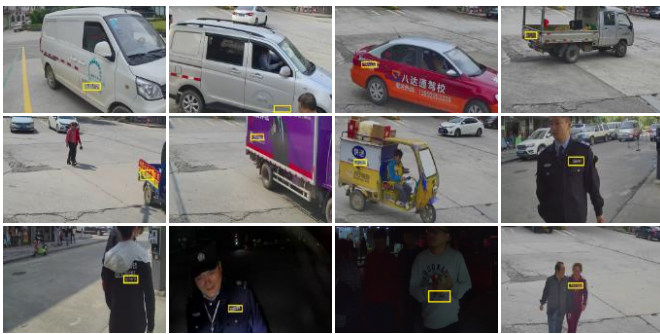


Figure 1. Examples of false-alarmed “license plates”

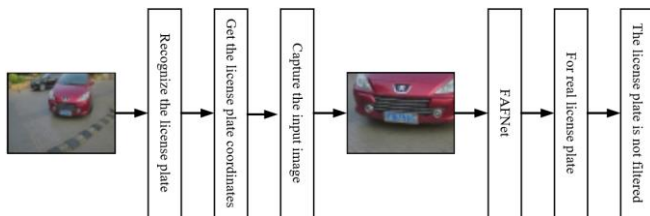


Figure 2. The process of false alarm filter of license plate recognition

The main contributions of this paper are summarized as follows:

(1) Aiming at the problem of false alarms in license plate recognition, this paper proposed a high-precision and real-time false alarm filter method;

(2) Aiming at the different sizes of license plates in different countries, this paper proposed a model generalization method to solve the non-universality problem of existing false alarm filter models;

(3) This paper built a large-scale sample dataset, the data samples contained in this dataset came from industries, and they covered various scenarios such as blurred images, poor lighting situations, cases under the influence of physical factors, and various weather conditions.

The rest of this paper is organized as follows: the second chapter elaborates on some work details of this research, including the cropping rule of input images, and the design of the model structure; the third chapter introduces the dataset and some details about the training, it also performs a series of experiments to demonstrate the powerful performance of FAFNet; the fourth chapter discusses solutions to the model generalization problem; the fifth chapter gives the conclusion of this study.

2. METHODOLOGY

This chapter introduces the content of three parts: the first part gives a brief introduction to the license plate detection algorithm adopted in this paper, the YOLOv5 network; the second part describes the cropping rule of the generated FAFNet input images and determines the parameters for image cropping; the third part gives design details of the FAFNet.

2.1 License plate detection algorithm

At first, this paper detected and located the license plate in the image. The YOLO target detection algorithm is one of the most advanced target detection networks at present, and YOLOv5 is a single-stage target detection algorithm. YOLOv5 adds some new improvements based on YOLOv4 [7], and its performance in both speed and accuracy have been greatly improved, therefore, this paper chose to use the YOLOv5 target detection algorithm to detect images.

2.2 Cropping rule

As mentioned above, the input images of FAFNet were obtained by cropping the original images based on the license plate coordinates and cropping parameters obtained from license plate detection. Moreover, in this paper, the car front features of real license plates were utilized to distinguish the real and false license plates, so the rule for cropping the input images according to the license plate coordinates is: for frontal vehicle images, the image cropped according to the license plate coordinates must be a complete car front image.

In this paper, assuming: m represents the ratio of width of the cropped image to the width of the license plate; n represents the ratio of the height of the cropped image to the height of the license plate; the center point of the cropped image is set to be the center point of the upper boundary of the license plate, so that more useful information about the car front could be obtained in the vertical direction, the calculation formulas are:

$$w_{crop} = m \times w_{lp} \quad (1)$$

$$h_{crop} = n \times h_{lp} \quad (2)$$

$$center_x_{crop} = center_x_{lp} \quad (3)$$

$$center_y_{crop} = center_y_{lp} - h_{lp}/2 \quad (4)$$

where, $crop$ represents the cropped image; lp represents the license plate; w , h , $center_x$, and $center_y$ respectively represent the width, height, and coordinates of the center.

According to the cropping rule and a lot of empirical tests, $m=3$ and $n=8$ are the most suitable cropping parameters for Chinese license plates. Figure 3 gives examples of partial image cropping. In Figure 3(a), the blue boxes represent the license plate coordinate position returned by the license plate recognition algorithm. In Figure 3(b), the green boxes represent the cropped area, and Figure 3(c) are the input images of FAFNet.

2.3 Design of FAFNet

With the development of deep neural networks, many classic classification networks have been proposed, such as the

earlier ones LeNet [8] and Alexnet [9], and the later ones, the more powerful VGGNet [10], GoogLeNet [11, 12], and ResNet [13], etc. Although these powerful networks can get good results on classification tasks, in terms of model size and recognition speed, they cannot meet the requirement for using them on embedded devices. Therefore, this paper made use of the existing network construction skills to design a new lightweight classification network.

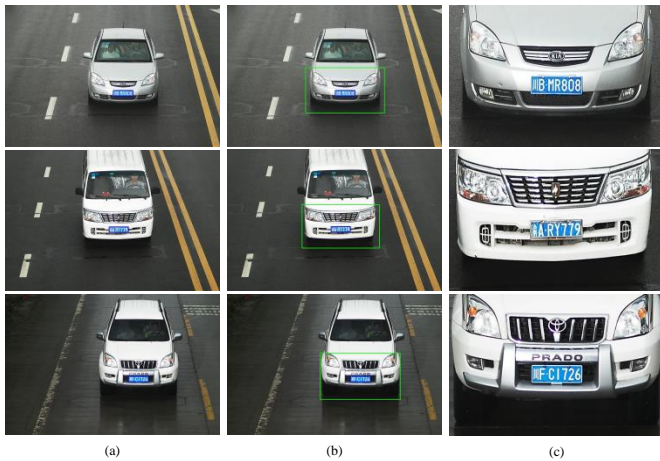


Figure 3. Examples of images cropped according to the cropping rule

The network architecture proposed in this paper was inspired by the DenseNet [14] image classification model. Compared with the above-mentioned traditional networks, DenseNet has fewer parameters, smaller model size, and faster calculation speed, it can meet the requirements of constructing a lightweight network. Moreover, its unique dense structure deepens the depth of the network, and at the same time enables the model to have the effects of implicit supervision and regularization, which can ensure the high accuracy of the model.

Dense block. Dense block is the main component of the proposed network, it is consisted of a series of dense cells. In this paper, the dense cells were defined as $[1 \times 1 \text{ conv}, 3 \times 3 \text{ conv}]$, wherein $s \times s \text{ conv}$ is a composite function (s is the size of the convolution kernel), which contains three parts: $s \times s$ convolution(conv), batch normalization (BN) [15], and the leaky rectified linear unit (Leaky ReLU). In the network, $1 \times 1 \text{ conv}$ is taken as the bottleneck layer, its position is before $3 \times 3 \text{ conv}$. Szegedy et al. pointed out that the bottleneck layer can reduce the number of input feature images, thereby improving the calculation efficiency [12, 13]. The specific structures of "conv", dense block, and dense cell are shown in Figure 4. In dense cells, the 1×1 convolution generates λk ($\lambda > 1$) feature images, and the 3×3 convolution outputs k feature images. If a dense block contains m dense cells, then it will generate mk feature images.

Transition layer. The transition layer is an important part of FAFNet. Its main function is to connect the dense blocks in the network, and to merge and downsample the features of the dense blocks. The transition layer includes a $1 \times 1 \text{ conv}$ layer and a 2×2 maximum pooling layer. The main function of $1 \times 1 \text{ conv}$ is to merge the concatenation features in the dense blocks, and at the same time increase the depth of the network, the number of its output images is the same as the output of the dense blocks; and the 2×2 maximum pooling layer is used for feature downsampling.

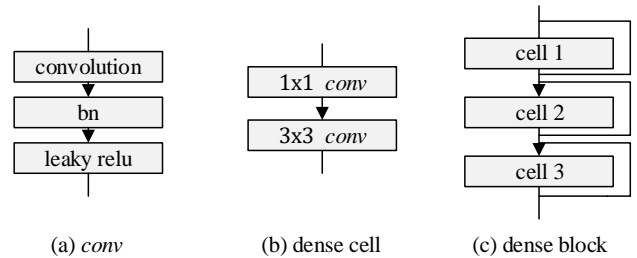


Figure 4. Specific structures of "conv", dense block, and dense cell

Implementation details. In this paper, the FAFNet has 4 dense blocks and 3 transition layers. Its input image size is $112 \times 112 \times 3$. Before the first dense block, this paper designed a conv layer to process the input images, the conv layer has a convolution kernel with a size of 3×3 , and its output has 8 channels. As for the 3×3 convolutional layer, its step was set to 1, and each side of the input was filled with 0 by a pixel to keep the size of the feature image unchanged. In this paper, a transition layer was set between two adjacent dense blocks, and the number of output channels was doubled after each transition layer. The number of dense cells in each dense block is $\{1, 2, 4, 4\}$. For each dense cell, it's set $k=8$ and $\lambda=2$ in this paper. After the last dense block, this paper designed a 1×1 convolutional layer and a linear activation layer, the number of output channels is equal to the number of classification labels; then the global average pooling layer was applied, and the softmax classifier was adopted for classification. The overall structure of FAFNet is shown in Table 1.

Table 1. Structure of FAFNet

Layer	Output size	FAFNet
Convolution	$112 \times 112 \times 8$	$3 \times 3 \text{ conv}$, stride = 1
Pooling	$56 \times 56 \times 8$	$2 \times 2 \text{ max}$, stride = 2
Dense Block 1	$56 \times 56 \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 1$
Transition Layer 1	$56 \times 56 \times 16$ $28 \times 28 \times 16$	$1 \times 1 \text{ conv}$, stride = 1 $2 \times 2 \text{ max}$, stride = 2
Dense Block 2	$28 \times 28 \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 2$
Transition Layer 2	$28 \times 28 \times 32$ $14 \times 14 \times 32$	$1 \times 1 \text{ conv}$, stride = 1 $2 \times 2 \text{ max}$, stride = 2
Dense Block 3	$14 \times 14 \times 64$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$
Transition Layer 3	$14 \times 14 \times 64$ $7 \times 7 \times 64$	$1 \times 1 \text{ conv}$, stride = 1 $2 \times 2 \text{ max}$, stride = 2
Dense Block 4	$7 \times 7 \times 96$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 4$
Convolution	$7 \times 7 \times 2$	$1 \times 1 \text{ conv}$, stride = 1
Classification	$1 \times 1 \times 2$	7×7 global average SoftMax

3. EXPERIMENTS AND RESULTS

This chapter first introduces the training of YOLOv5 and the test results, and gives the training data and training details of FAFNet; then, it compares FAFNet with the classic classification networks LeNet and AlexNet, and demonstrates the powerful performance of FAFNet in accuracy and speed on the dataset; At last, it discusses the influence of license plate, car front color, and edge binary image on the FAFNet model, and explores the main features extracted by the FAFNet model and the robustness of the model.

The experimental environment of this study was: Ubuntu 16.04 system, CPU Intel(R) Xeon(R) CPU E5-2603 v4 @ 1.70GHz, GPU NVIDIA GTX 1080Ti, and at the same time, the running speed of different models was tested on the embedded device HISI3516 AV200.

3.1 Training of YOLOv5 and the test results

In this paper, the YOLOv5 target detection algorithm was adopted for license plate detection. The training dataset of the algorithm came from related security departments. This paper

selected 128 images with license plates in different scenarios and manually labelled them. During the training, the fine-tuning method was applied in data training based on the YOLOv5 model, and Figure 5 gives the process training curves, the accuracy, and the recall rate. As can be seen from the figures, in the training process, the curves of loss decline steadily, while the curves of accuracy and recall rate grow steadily, indicating that the YOLOv5 model can well learn the features of the license plate images, and in different IoU thresholds (from 0.5 to 0.95, step length was 0.05), the average mAP reached 0.85.

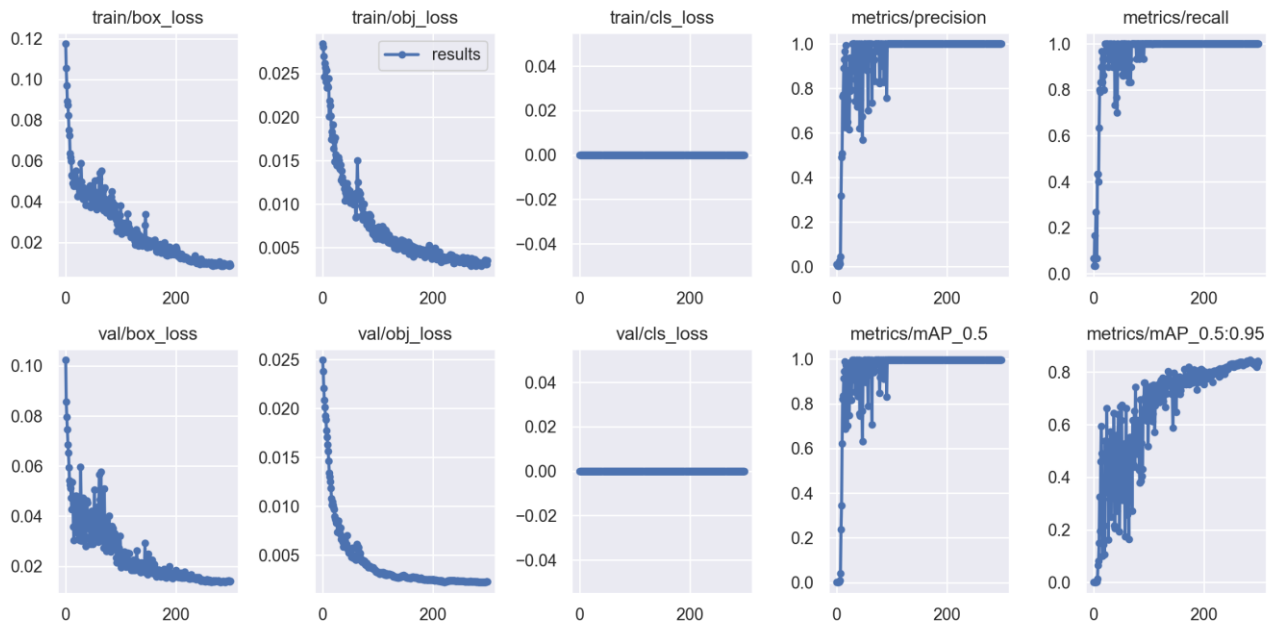


Figure 5. Curves of loss, accuracy, and recall rate of the training process of YOLOv5

3.2 Dataset and preprocessing

All data in the dataset built in this paper came from images collected by different security and surveillance cameras. The positive samples included the license plate recognition scenarios of different vehicle models during the day and night, and some special scenarios such as high noise, poor lighting, as well as rain and fog weathers, etc. The negative samples consisted of three parts. The first part is natural scenes without vehicles, such as green belts, road speed bumps, and road railings; the second part is the false alarm data collected by cameras, such as the characters on the car body, and texts in other natural scenes; the third part is some natural scene text images in the COCO-Text dataset [16]. In order to further expand the data volume of negative samples and increase the diversity of data, this paper carried out sample augmentation on negative samples, and the main operations included random rotation, size scaling, and random cropping.

The current dataset has a total of 215,675 images, of which 117,255 are positive samples and 98,420 are negative samples. The size of all images had been adjusted to 112×112 when being input into the network.

3.3 Training details of FAFNet

This paper selected 90% of the dataset as the training set and 10% as the test set. In the training process, stochastic gradient descent (SGD) was employed to train the FAFNet

model, the batch size of the training was set to 128, the initial learning rate was set to 10^{-3} , and the gradient weight (momentum) [17] of last time was set to 0.9; referring to [18], the weight decay term was set to 5×10^{-4} , the maximum number of training iterations was 105w, and the learning rate was multiplied by 0.1 at the 20w-th and the 60w-th iterations. Moreover, all training experiments in this research were completed in the Darknet framework [5].

3.4 Comparison of the results of FAFNet, LeNet, and AlexNet

To demonstrate the performance of the proposed algorithm and feasibility of deploying it on embedded devices, this paper selected the classic classification networks LeNet [8] and AlexNet [9] for comparison, and their experimental results on the dataset were compared, as shown in Table 2.

It can be seen from Table 2 that the FAFNet has a powerful performance. In terms of model size, the model size of FAFNet is only 0.11MB, which is 542 times and 1036 times smaller than the model size of LeNet and AlexNet respectively. In terms of recognition rate, FAFNet is slightly higher than AlexNet, and it is 0.14% higher than LeNet, which indicates that FAFNet has effectively avoided network parameter redundancy and greatly improved parameter utilization. In terms of floating-point operation operand, the operation operand it took for the FAFNet to complete one-time forward propagation is only 0.024GFlops, which is far lower than

LeNet and AlexNet. In terms of recognition speed, this paper compared the recognition speed of FAFNet on GPU, CPU and HISI; according to the table, on GPU, LeNet has a faster recognition speed due to its simple network structure, while the recognition speeds of AlexNet and FAFNet are alike; but on CPU and HISI, apparently the recognition speed of FAFNet is much faster than that of LeNet and AlexNet, and it can

realize real-time false alarm processing on HISI. The model size of FAFNet is very small, which can effectively reduce the memory occupation of the embedded devices; at the same time, its recognition speed can meet the real-time requirements of running on embedded devices, therefore, the FAFNet model is very suitable for embedded devices.

Table 2. Comparison of results with LeNet and AlexNet

Network	Model size (MB)	Recognition rate (%)	Floating-point operation operand (GFlops)	Recognition speed (ms/per image)		
				GPU	CPU	HISI
LeNet	59.7	99.75	0.16	1.5	60	250
AlexNet	114	99.98	0.479	2.2	170	1000
FAFNet	0.11	99.99	0.024	2.7	14	55

3.5 The influence of license plate area, car front color, and edge binary image on the model

This paper designed three experiments to explore the influence of license plate area, car front color, and edge binary image on the model, and to further verify the robustness of the

FAFNet model, as well as the features of positive samples extracted by the model. The training data of the experiments were a subset of the industrial dataset mentioned in Section 3.1. The corresponding filled image, grey image, and edge binary image are shown in Figure 6, and the experimental results are shown in Table 3.

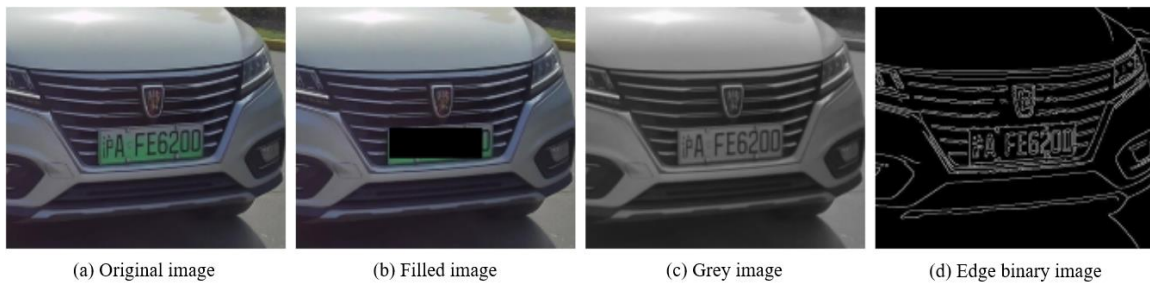


Figure 6. The input image of FAFNet and its corresponding filled image, grey image, and binary image

Table 3. Comparison results of original image, filled image, grey image, and edge binary image

Training image	Recognition rate (%)	Floating-point operation operand (GFlops)
Original color image	99.98	0.024
Filled image	99.97	0.024
Grey image	99.98	0.021
Edge binary image	99.94	0.021

License plate area. According to above analysis, the input images of the network were obtained based on the license plate coordinates and cropping parameters. The input images contained the information of the license plate area. Therefore, it's necessary to consider whether the information of the license plate area is also one of the main features extracted by the FAFNet model. If the license plate area information has a direct influence on the FAFNet model, then due to the diversity of license plates, the FAFNet model can hardly be applied to real scene scenarios.

When acquiring the input images, this study filled the license plate information with pixel 0, the license plate area information was completely shielded, and then the images were input into the network for training. By comparing the training results of unfilled and filled images of license plates, the influence of the features of license plate area on FAFNet could be analyzed. According to the experimental results shown in Table 3, the recognition rate after filling the license plate images is not much different from the recognition rate before image filling, therefore, what the FAFNet extracts is the information of the car front, and the license plate area basically has no influence on the model.

Car front color. According to above results, FAFNet mainly extracts the car front information. In reality, vehicles are of diverse and complex colors, so the diversity of car front colors is a big challenge to the generalization ability of the false alarm filter model. In order to verify the influence of car front color information on the model, this paper converted the color input images into the grayscale images to mask the car front color information. It can be seen from the experimental data that the model maintained a strong false alarm filter ability after the color information of car front had been shielded, indicating that the model is very robust to the color information of car front.

Edge binary image. In order to further verify the main features extracted by the FAFNet model, this paper converted the network input images into edge binary images, and only edge information of the images had been retained. First, the input images were subjected to mean filtering, then the edge binary images were attained by the Canny edge detection algorithm [19] and the Otsu threshold method (Ostu) [20]. After that, the edge binary images were taken as training data and input into the network for training, and the results are given in Table 3. As can be seen from the results, the effect of

false alarm filter of the model based on edge binary image is not much different from that of the model based on color image, therefore, it can be concluded that what is mainly extracted by the FAFNet model is the edge contour features of the car front.

Based on above three experiments, we can know that the license plate area information and car front color information have little influence on the model. As for the model based on edge binary image, if the edge detection and binarization algorithm could be further optimized, then the model should have a better effect than the model based on color image. At the same time, the traffic data of real scene scenarios adopted in this paper had covered complex situations, which can further prove that the proposed algorithm is highly robust.

4. MODEL GENERALIZATION

In this paper, the FAFNet model had obtained excellent results in the false alarm filter task for Chinese license plates. However, we wish that the FAFNet could be used to solve the problem of false alarm filter of the license plates of other countries. To this end, this paper proposed two solutions: one is to collect positive and negative sample data and train the false alarm filter model for the false alarm filter tasks of different countries; the other is to use some preprocessing methods to make the false alarm filter model designed for Chinese license plates be applicable for the situations of other countries. This paper chose the second solution to solve the model generalization problem.

When the Chinese license plate false alarm filter model was applied to the false alarm filter tasks of other countries, the obtained effect was quite unsatisfactory. After analysis, it is found that the main reason is the different sizes of license plates in different countries. When the input images were cropped according to the cropping rule mentioned in Section 2.1, the cropped images could not be well applied to the current FAFNet model. Therefore, the key to solving model generalization problem is how to crop proper input images suitable for the FAFNet model according to the coordinates of the license plates of other countries.

After counting the identified license plate coordinates of other countries (special license plates hadn't been taken into consideration, such as the double-layer license plate), it is found that the height of the license plates is almost the same. Therefore, for the license plates of other countries, this paper considered calculating an area similar to the size of Chinese license plates based on the actual license plate coordinates, and then cropped the input images based on this area, and this area is called the "virtual license plate".

According to the coordinates of the identified foreign license plates, this paper set its height as h_a , width as c , and the aspect ratio as $ratio_a$, then, the height of the virtual license plate h_v could be calculated:

$$h_v = \begin{cases} h_a & ratio_a > 1.5 \\ \theta \times h_a & ratio_a \leq 1.5 \end{cases} \quad (5)$$

where, it's defined that, when $ratio_a > 1.5$, the license plate is a single-layer license plate, at this time, the height of the virtual license plate is equal to the height of a real license plate; when $ratio_a < 1.5$, the license plate is a double-layer license plate, and the height of the virtual license plate is θ times ($\theta < 1$) the height of a real license plate. After statistics and tests, this paper set

$\theta = 0.6$.

The width of the virtual license plate w_v is:

$$w_v = ratio_{base} \times h_v \quad (6)$$

where, $ratio_{base}$ is the statistical value of the aspect ratio of Chinese license plates, and $ratio_{base} = 4$.

According to Formulas 5 and 6, the coordinates of the virtual license plate could be calculated, then, according to the cropping rule proposed, proper images could be cropped and input into the false alarm filter model.

This paper realized the generalization of the false alarm filter model based on the second solution, compared with the first solution, it has the following two advantages:

- (1) It reduced the cost of data collection, since it doesn't need to collect the data of license plates of other countries;
- (2) It reduced the cost of training, since it doesn't need to collect the data of license plates of each country.

Therefore, it is only necessary to train the false alarm filter model of one country, and then through the method of virtual license plate, it can be applied to other countries.

5. CONCLUSION

Aiming at the possible detection errors of the current license plate location technology when carrying out the license plate recognition tasks, this paper proposed a powerful license plate recognition false alarm filter algorithm: the FAFNet, a lightweight neural network built based on deep CNN and is capable of end-to-end training. In this paper, at first, the design details of the FAFNet were given, the unique network structure of FAFNet not only greatly reduced the number of parameters, increased the calculation speed, but also deepened the network depth, thereby the model could reach a high accuracy. Then, aiming at the generalization problem of the false alarm filter model, this paper also proposed a virtual license plate method which makes the proposed model be applicable to other countries, while avoiding repeated data collection and model training. At last, through experiments on industrial dataset, the FAFNet's excellent false alarm filtering performance and recognition speed on different hardware devices (GPU, CPU, HISI) had been verified, and the results proved that FAFNet is fully applicable to embedded devices and can achieve efficient and real-time filtering function.

REFERENCES

- [1] Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V., Kayafas, E. (2008). License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 9(3): 377-391. <https://doi.org/10.1109/TITS.2008.922938>
- [2] Li, H., Shen, C. (2016). Reading car license plates using deep convolutional neural networks and LSTMs. arXiv preprint arXiv:1601.05610.
- [3] Li, H., Shen, C. (2016). Reading car license plates using deep convolutional neural networks and LSTMs. arXiv preprint arXiv:1601.05610.
- [4] Jain, V., Sasindran, Z., Rajagopal, A., Biswas, S., Bharadwaj, H.S., Ramakrishnan, K.R. (2016). Deep automatic license plate recognition system. *Proceedings*

- of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, Guwahati Assam, India, pp. 1-8. <https://doi.org/10.1145/3009977.3010052>
- [5] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25(2): 1097-1105. <http://dx.doi.org/10.1145/3065386>
- [6] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision*, Amsterdam, The Netherlands, pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
- [8] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1): 62-66. <https://doi.org/10.1109/TSMC.1979.4310076>
- [9] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278-2324. <https://doi.org/10.1109/5.726791>
- [10] Zherzdev, S., Gruzdev, A. (2018). Lprnet: License plate recognition via deep neural networks. *arXiv preprint arXiv:1806.10447*.
- [11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Li, F. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
- [12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, pp. 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [13] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2818-2826. <https://doi.org/10.1109/CVPR.2016.308>
- [14] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [15] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 4700-4708. <https://doi.org/10.1109/CVPR.2017.243>
- [16] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, Lille, France, pp. 448-456.
- [17] Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.
- [18] Sutskever, I., Martens, J., Dahl, G., Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *International conference on machine learning*, Atlanta, Georgia, USA, pp. 1139-1147.
- [19] Xuan, L., Hong, Z. (2017). An improved canny edge detection algorithm. *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, pp. 275-278. <https://doi.org/10.1109/ICSESS.2017.8342913>
- [20] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6): 679-698. <https://doi.org/10.1109/TPAMI.1986.4767851>