Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6591873, 10 pages
https://doi.org/10.1155/2020/6591873

Research Article

Vehicle Logo Recognition with Small Sample Problem in Complex Scene Based on Data Augmentation

Xiao Ke [1,2,3] **and Pengqiang Du** [1,2,3]

[1]*College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China*
[2]*Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing (Fuzhou University), Fuzhou 350116, China*
[3]*Key Laboratory of Spatial Data Mining & Information Sharing, Ministry of Education, Fuzhou 350003, China*

Correspondence should be addressed to Pengqiang Du; fzu_pengqiangdu@outlook.com

Received 1 April 2020; Revised 22 May 2020; Accepted 9 June 2020; Published 9 July 2020

Guest Editor: Feng-Jang Hwang

Automatic identification for vehicles is an important topic in the field of Intelligent Transportation Systems (ITS), and the vehicle logo is one of the most important characteristics of a vehicle. Therefore, vehicle logo detection and recognition are important research topics. Because of the problems that the area of a vehicle logo is too small to be detected and the dataset is too small to train for complex scenes, considering the speed of recognition and the robustness to complex scenes, we use deep learning methods which are based on data optimization for vehicle logo in complex scenes. We propose three augmentation strategies for vehicle logo data: cross-sliding segmentation method, small frame method, and Gaussian Distribution Segmentation method. For the problem of small sample size, we use cross-sliding segmentation method, which can effectively increase the amount of data without changing the aspect ratio of the original vehicle logo image. To expand the area of the logos in the images, we develop the small frame method which improves the detection results of the small area vehicle logos. In order to enrich the position diversity of vehicle logo in the image, we propose Gaussian Distribution Segmentation method, and the result shows that this method is very effective. The $F1$ value of our method in the YOLO framework is 0.7765, and the precision is greatly improved to 0.9295. In the Faster R-CNN framework, the $F1$ value of our method is 0.7799, which is also better than before. The results of experiments show that the above optimization methods can better represent the features of the vehicle logos than the traditional method, and the experimental results have been improved.

1. Introduction

The core idea of intelligent traffic [1] is to use sensors, cameras, and other ways to collect traffic data and use computer-aided management to replace the traditional manual monitoring. For example, we can track moving objects in the video [2, 3] and analyze the salient objects [4, 5] to find traffic abnormalities. It can quickly complete traffic data analysis [6], sharing and retrieval, to achieve the integration of traffic management. Computer vision technology plays a vital role in this field. For a vehicle, its license plate and logos are two important pieces of information. There are various researches on license plate recognition [7, 8], and numerous mature and stable systems have been developed and widely used in actual production scenarios. On the contrary, the research of vehicle logo recognition is not enough. And this is an important part of intelligent transportation; therefore, the demand for vehicle recognition will be increasing in the future. There have been some studies on the recognition of the vehicle logo. Yu et al. [9] proposed a Bag-of-Words-based method for vehicle logo classification. Firstly, dense-SIFT (dense-SIFT) feature extraction is performed on the vehicle logo. After feature extraction, these features are divided into k clusters. Then, the histograms of each image corresponding to these clusters are counted and described. The feature of the whole image is finally trained and classified by support vector machine (SVM). Cyganek et al. [10] put forward a vehicle recognition method based on Tensor Representation in literature. This

method establishes a corresponding 4D tensor classifier for each class of vehicle logos and achieves the function of classifying each vehicle logo by combining the established classifiers. However, this method also locates the vehicle logo by detecting the prior position of the license plate. Llorca et al. [11] attempted to use a variety of features for the vehicle logo and finally decided to construct the system in the form of gradient histogram (HOG) features and support vector machine (SVM). Peng et al. [12] proposed a statistical random sparse distribution method to extract the features of the vehicle logo and then classify the extracted features by the classifier. This method is optimized for the vehicle logo, so it is more robust than some general features such as HOG and SIFT. Huang et al. [13] proposed a vehicle recognition system based on pretraining strategy. Detection and recognition are achieved by combining traditional detection methods with the recognition of convolutional neural networks. In order to reduce the false detection error caused by the traditional detection method, the system expands the detection range. Soon et al. [14] proposed a new pretreatment method. Before using the deep neural network method to extract features, they use the white transformation technology to process the image, removing the pixel redundancy of the adjacent image, making the characteristics of the vehicle logos more obvious.

In previous studies, many methods of vehicle logo classification have been proposed [15], but the location of the vehicle logo has always been a difficult problem in Intelligent Transportation Systems. In the conventional method, the location needs some prehypothesis, such as using the prior position of the license plate or using the characteristics of texture features of headlights to locate the vehicle logo [16], which is not robust to complex situations. However, at present, many deep networks have been able to automatically learn the features of the vehicle logo from the image, thereby directly detecting and recognizing the vehicle logos. But there are also some problems in the application of deep neural networks. The parameters of deep neural networks are very large. At the same time, because of the complexity of its construction, there are many hyperparameters, so it is very difficult to optimize them [17]. When the size of the dataset is too small, the performance of the network will be worse. In order to make the deep neural network better learn and extract the feature representation, we adopt a variety of data optimization methods to make the network perform better in the data of vehicle logos.

In our research, we put forward a new research idea. Compared with the traditional methods, we propose a data optimization method based on the combination of multiple strategies, which is based on the location diversity and scale diversity of vehicle logo.

2. Related Work

The method proposed in this paper is based on the existing deep learning method. Therefore, our research needs a network framework with good performance. We choose the two most representative frameworks, YOLO and Faster R-CNN, as our research objects. At the same time, we need Nonmaximum Suppression algorithm to optimize the detection results.

2.1. YOLO. YOLO algorithm is one of the most popular target detection algorithms [18–20]. Its main advantage is that it can achieve real-time detection, but also get better detection results. The main idea of YOLO algorithm is to divide the image and then directly carry out regression detection on the divided area. It realizes end-to-end detection by inputting images directly into the network to predict. Compared with the existing target detection system, YOLO has a general performance in locating the target area, but it has a better inhibition on the false-positive area of the background.

YOLO detects objects by dividing the image into grids. For the object in the image, when its center is in a grid, the grid is responsible for detecting the object. For these grids, each grid predicts several rectangular boxes that may contain objects, each of which corresponds to a confidence score. This confidence reflects the quality of the prediction of objects for each bounding box. The higher the score, the closer the box will be to the ground truth. If the rectangular box does not contain objects, the confidence should be zero. For each rectangular box, there are five parameters after the prediction: the value of the X and Y coordinates of the object center relative to the grid, the ratio of the rectangular length to the width of the image, and the confidence score mentioned above. In addition, each grid also predicts several conditional probabilities. These probabilities are used to determine the categories of objects, and the number is equal to the number of categories in the dataset.

For the experiment in this paper, five boxes are predicted in each grid. Each box contains four coordinate numbers, one confidence and 30 category probabilities, so each grid has $5 * (5 + 20) = 175$ output dimensions.

2.2. Faster R-CNN. Before the advent of Faster R-CNN, the target detection algorithms proposed by researchers need to use time-consuming region recommendation algorithms to infer the target region [21, 22]. Although some researchers have taken various optimizations to the algorithm, the time consuming of the algorithm is still very huge. In this context, researchers put forward the idea of using CNN directly to predict the target area, and RPN algorithm was born [23].

Faster R-CNN consists of four parts: Conv layer, RPN, ROI Pooling, and Classifier. Conv layer is a convolutional neural network, which is mainly used to extract the features of the original image. It can use VGG [24] or ResNet [25]. The extracted feature map will be used for the RPN layer and full connection layer. RPN is the core idea of Faster R-CNN. It is a full convolution network used to generate region proposals. Firstly, the feature image obtained by Conv layer is further convoluted, and the anchors are generated for each pixel on the image. After that, these anchors are classified by softmax function to determine whether they are the target area. At the same time, anchors are modified by bounding box regression to get more accurate proposals. The Roi Pooling layer is relatively simple. It uses the feature map from Conv layer and the proposals from RPN to form a fixed-size proposal feature map which is then sent to the full connection layer for target recognition and location. Finally, the Classifier sends the proposed feature map from the Roi Pooling layer to the fully

connected network and uses the softmax function to classify. At the same time, L1 loss is used to complete the bounding box regression to obtain the exact position of the object.

2.3. Nonmaximum Suppression. Nonmaximum Suppression [26] (NMS) is very important in the field of computer vision. At present, many mature technologies, such as face detection [27] and pedestrian detection [28], are applied. In the process of object detection, multiple rectangular boxes are often generated. How to filter these candidate boxes to best make the final detection effect has become a problem, and the NMS can be used to complete this task.

The core idea is to compare the confidence of the two candidate boxes and discard the smaller part when the overlap of the two candidate boxes (IOU) is greater than a certain threshold. Loop like this until the IOU of any two candidate boxes is less than the threshold value. At this point, the confidence of the remaining candidate box is the highest, and the detection effect is the best.

3. Proposed Method

3.1. Augmentation and Its Significance. In the field of deep learning, big data is the basis to support the learning of the features of objects [29]. Training a network requires a large amount of data as a support to better extract the features of the targets. If the data quality in a dataset is not good enough, data equilibrium [30] and data augmentation are usually used to optimize it. The significance of data augmentation is to transform the training data and generate new data by certain methods. Through data augmentation, the original dataset can be optimized and expanded. It can prevent the overfitting [31] caused by the small amount of data in the training process. It is of great significance to the recognition and detection ability of the model. For the current vehicle logo datasets, most of them have separated the logos from the scenes, used for classification of vehicle logo images. This kind of dataset cannot be used for training. However, the scale of the dataset labelled directly on the original image is too small to meet the amount of data needed to train. Therefore, data augmentation should be performed on the dataset to meet the needs.

3.2. Traditional Data Augmentation Methods

3.2.1. Rotation. The pixels in the image rotate randomly around the center at the same angle.

3.2.2. Flipping. The pixels symmetrical with horizontal or vertical centerlines are exchanged, which causes the entire image to be upside down or left to right.

3.2.3. Brightness. The overall brightness of the image is promoted or reduced.

3.2.4. Contrast. The difference between the brightest pixel and the darkest pixel in the image is enlarged or narrowed, and the pixel values between them are transformed accordingly.

3.2.5. Noise. The RGB channel value of each pixel in the image is randomly transformed in a proportional, and noise is introduced to cause the image to change.

3.2.6. Cropping. The upper, lower, left, and right boundaries of the image plane are cut with a certain proportion or a certain pixel width.

Many augmentation methods have been proposed so far, but not all augmentation methods are applicable to the vehicle logo dataset. The augmentation methods described above can be broadly divided into two categories: pixel-level operations on the original RGB channel such as brightness transformation and noise disturbance, and no pixel-level operations such as cropping transformation. Through the use of these data augmentation methods, the image becomes more diversified, so that the network can learn the image feature representation more fully.

3.3. Cross-Sliding Segmentation Method. For convolutional neural networks, because of the huge parameters, the amount of data needed is also very large, so we need to find a simple and fast way to generate a large number of effective data. Because the logo area is generally small, it is not easy to segment the logo area when the image is segmented. At the same time, because the location of vehicle logo is very variable, cross-sliding segmentation is used to quickly increase the number of effective images. This method is very fast and easy to implement.

We scan and intercept the image using a rectangular box with a length and width of 1/2 of the source image as Figure 1. This method greatly increases the amount of data. Because the area of the logo is small, it is not easy to split the ground truth boxes, which results in invalid data. The specific segmentation steps are as follows.

Firstly, the original image is divided into 9 subimages, and the length and width of each subimage are 1/2 of the original image:

$$\begin{cases} nw = \frac{1}{2} * ow, \\ nh = \frac{1}{2} * oh. \end{cases} \tag{1}$$

The position of the generated image in the original image is

$$\begin{cases} xx1 = \frac{1}{4} * nw * i, \\ yy1 = \frac{1}{4} * nh * j, \\ xx2 = xx1 + nw, \\ yy2 = xx2 + nh. \end{cases} \tag{2}$$

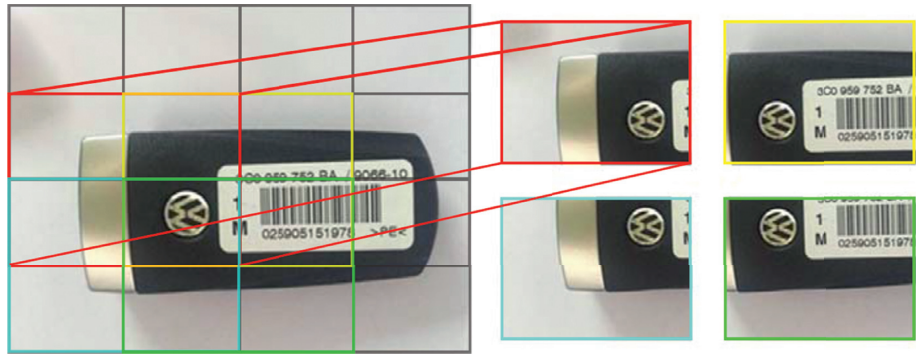

Figure 1: The image to be split. The image above is divided into nine, four of which contain the complete target area.

Among them, $i = 0, 1, 2$; $j = 0, 1, 2$; $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; nw and nh are the width and height of the generated image, respectively; ow and oh are the width and height of the original image.

All the generated subimages form set u, while the images that can be used for training form set A. If the target area is r, the subimage that can be used for training should contain r, so

$$A = \{u \mid u \subseteq U, r \subseteq u\}. \tag{3}$$

Cross-sliding segmentation method produced a total of 9 images on Figure 1, 4 of which contain complete vehicle logo information for training, and the remaining should be discarded directly because they do not contain the vehicle logo information or the vehicle logo information is incomplete.

3.4. Small Frame Method. In general images, the proportion of vehicle logos in the whole image is very small, and the background information is too complex, which leads to the lack of main information. At the same time, because the background information is too much, the deep neural network will encounter a lot of noise in the learning process, causing the convergence speed to be too slow.

Therefore, how to suppress the background and highlight the characteristics of the logo has become a difficult problem. In order to make the vehicle logo contribute more information to the whole image, it is necessary to remove the area without vehicle logo. A very easy and effective way is to directly expand the proportion of the logo size in the image.

In this paper, a small rectangular box of random size is used to select the vehicle logo, and other areas without the vehicle logo are discarded to highlight the vehicle logo, so that the network can better extract the features of the vehicle logos.

The position of the generated image in the original image is

$$\begin{cases} xx1 = x1 - \text{random}(1.5,\ 3) * rw, \\ yy1 = y1 - \text{random}(1.5,\ 3) * rh, \\ xx2 = x2 + \text{random}(1.5,\ 3) * rw, \\ yy2 = y2 + \text{random}(1.5,\ 3) * rh. \end{cases} \tag{4}$$

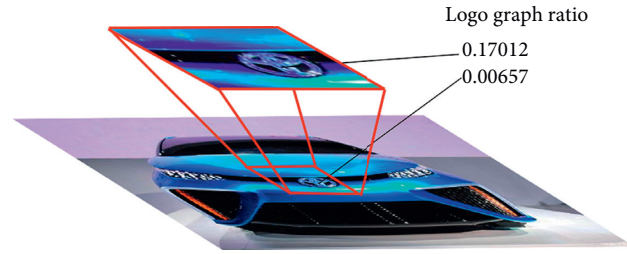

Figure 2: There is an image processed by the small frame method. The vehicle logo accounts for a larger proportion of the graph after processing.

Among them, $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; $x1$ and $y1$ are the positions of the upper left corner of the target area in the original image; $x2$ and $y2$ are the positions of the target area in the lower right corner of the original image; the function random $(1.5,\ 3)$ is a function that randomly generates a random number between 1.5 and 3; rw and rh are the width and height of the target area.

Definition 1. The logo graph ratio is the ratio of the area of the logo ground truth to the area of the whole image.

The average graph ratio can measure the situation of the target region in the overall image dataset. The vehicle logo is a small target in the field of object detection, and its graph ratio is generally low:

$$\text{avgRatio} = \frac{1}{N} \sum_{i}^{N} \frac{\text{Area}(ci)}{\text{Area}(oi)}. \tag{5}$$

Among them, avgRatio is the average of all logo graph ratios and Area (ci) is the area size of the logo in image I and Area (oi) is the area size of image I.

Using the method mentioned above, the logo graph ratio of Figure 2 increases from 0.00657 to 0.17012, and the average logo graph ratio of enhanced dataset increases from 0.0144 to 0.0526, about 3.65 times.

3.5. Gaussian Distribution Segmentation Method. Because of the existence of pooling operations in convolutional neural networks, the robustness of the networks to data translation

is very poor. Some studies have shown that, for the same image and the same network, as long as the input image is slightly modified or shifted by one pixel, the output of CNN will change dramatically, and the deeper the network layer is, the more likely this error will occur. The researchers think that there is a certain photographic bias in the general dataset. On the macro level, as long as it is not pixel-level coding, there are no two identical images in the world, so the neural network cannot learn the strict translation invariance and does not need to learn.

The position of the vehicle logo in the image is very variable, so there is a high demand for the diversity of the position of the target in the training set. When the dataset is small, we must optimize the location of the vehicle logo in the dataset. Therefore, we propose the Gaussian Distribution Segmentation method. After the dataset is optimized by this method, the position of the target region will be Gaussian distribution. The size of the generated image is

$$\begin{cases} nw = \max(ox,\ ow - ox), \\ nh = \max(oy, oh - oy). \end{cases} \tag{6}$$

Among them, nw and nh are the width and height of the generated image; ox and oy are the X coordinate and Y coordinate of the target region center of the original image; ow and oh are the width and height of the original image.

We need to build a function and the function is used to generate a number between 0 and 1, which should meet the standard normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{(-x^2/2)}. \tag{7}$$

The coordinates of the target center point of the generated image are

$$\begin{cases} nx = \dfrac{ow}{2} + (nw - rw) * \text{randn}(0, 1), \\ ny = \dfrac{oh}{2} + (nh - rh) * \text{randn}(0, 1). \end{cases} \tag{8}$$

Among them, nx and ny are the X and Y coordinates of the target center point of the generated image; ow and oh are the width and height of the original image; nw and nh are the width and height of the generated image; rw and rh are the width and height of the target rectangle; and randn $(0, 1)$ is a function that generates a number between 0 and 1 satisfying the Gaussian distribution.

The position of the generated image in the original image is

$$\begin{cases} xx1 = ox - nx, \\ yy1 = oy - ny, \\ xx2 = xx1 + nw, \\ yy2 = xx2 + nh, \end{cases} \tag{9}$$

where $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; ox and oy are the X coordinate and Y coordinate of the target center of the original image; nx and ny are the X and Y coordinate of the target center of the generated image; nw and nh are the width and height of the generated image.

4. Results and Discussion

4.1. Experiment Setup. The data in this paper are from the Big Data and Computational Intelligence Competition (BDCI). There are 1131 images of different sizes with annotations totaling 30 categories. And we use 1006 images in its test dataset as the test set of this experiment. The distribution of classes is shown in Figure 3.

Because of the need for data quantity, we augment the original data to expand the dataset. As a contrast, we have experimented with a variety of traditional augmentation methods and the methods proposed in this paper. We try our best to control the amount of data generated by various methods at the same level to ensure the accuracy of the experiment. For these methods, we divide them into two categories: pixel-level operation and non-pixel-level operation. Among them, pixel-level operations include brightness transformation, contrast transformation, and noise addition. Non-pixel operations include cropping, rotation, and flipping. Our own methods based on logo data optimization are all non-pixel-level methods. The amount of data generated by all methods is 2-3 times that of the original dataset. The data distribution generated by these methods is shown in Figure 4.

4.2. Experimental Results. In this paper, three parameters, P, R, and $F1$, are used as the criteria for judging and evaluating the experimental process. Their calculation methods are as follows:

$$\begin{aligned} R &= \frac{1}{N} \sum_{i}^{N} \frac{\text{Correct}(i)}{\text{Ground}(i)}, \\ P &= \frac{1}{N} \sum_{i}^{N} \frac{\text{Correct}(i)}{\text{Propersal}(i)}, \\ F1 &= \frac{2 * P * R}{P + R}. \end{aligned} \tag{10}$$

Among them, Correct (i) is the number of logos correctly detected, Propersal (i) is the total number of logos detected, and Ground (i) is the actual number of logos.

We study the performance of various data optimization methods under the framework of YOLO. By adjusting the parameters and the number of iterations, we get the optimal results under the YOLO framework. First of all, the original data is tested, and the loss changes and data results are as shown in Figure 5.

We trained 50000 iterations under the framework of YOLO and output the results every 1000 iterations. Observing the experimental results, we can find that there is a sudden drop in the loss value between the 10000 and 20000 iterations. The reason is that in this place we use the strategy of learning rate decay, which can make the loss worth further decline. At the same time, it can be seen from (b) that $F1$ value tends to be stable before 50000 iterations, so we can get

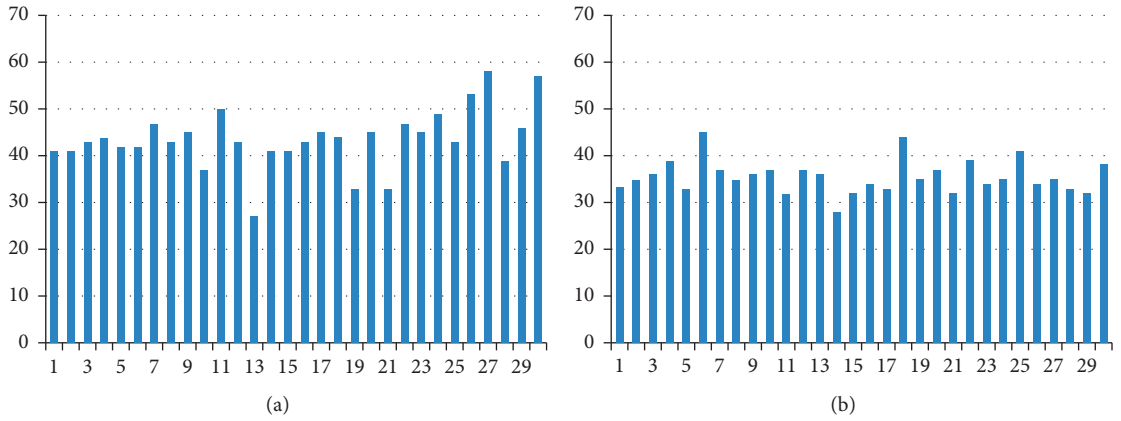

Figure 3: Image distribution of each class. (a) Distribution of training samples. (b) Distribution of test samples.

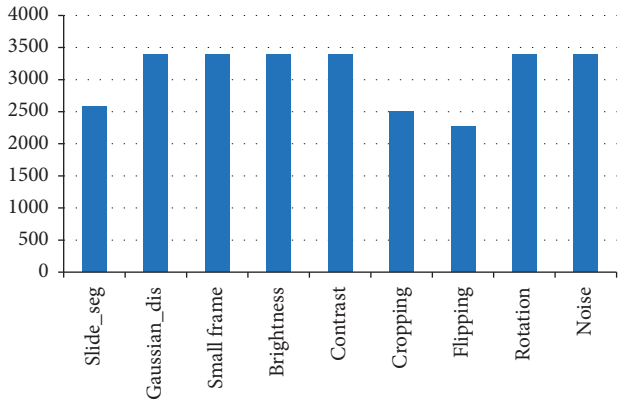

Figure 4: The distribution of new images. By data augmentation, the size of the dataset is expanded to 28859.

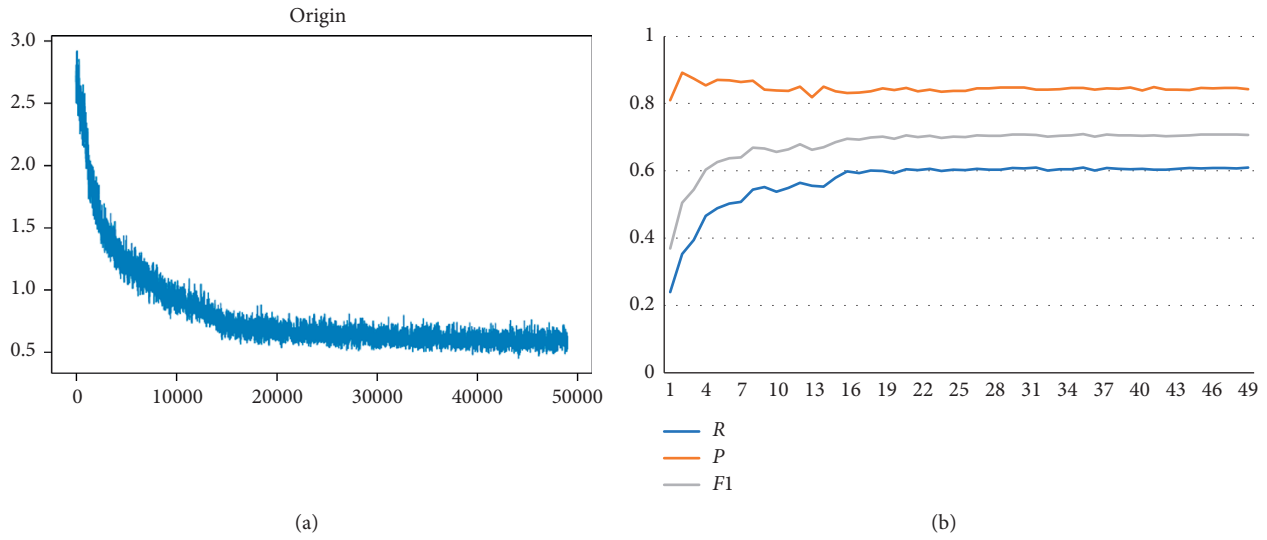

Figure 5: Results of original experiment. We can see that the loss value is stable after 20000 iterations. (a) Loss curve. (b) Curve of R value, P value, and $F1$ value.

an optimal model in 50000 iterations of training. The best results of training on the original dataset are shown in Table 1.

After that, we tried to add the optimized data to the network for training. We experimented with nine enhancement methods, and their loss values changed as shown in Figure 6.

Table 1: The best detection result of the original dataset on YOLO.

P	R	$F1$
0.8466	0.6096	0.7088

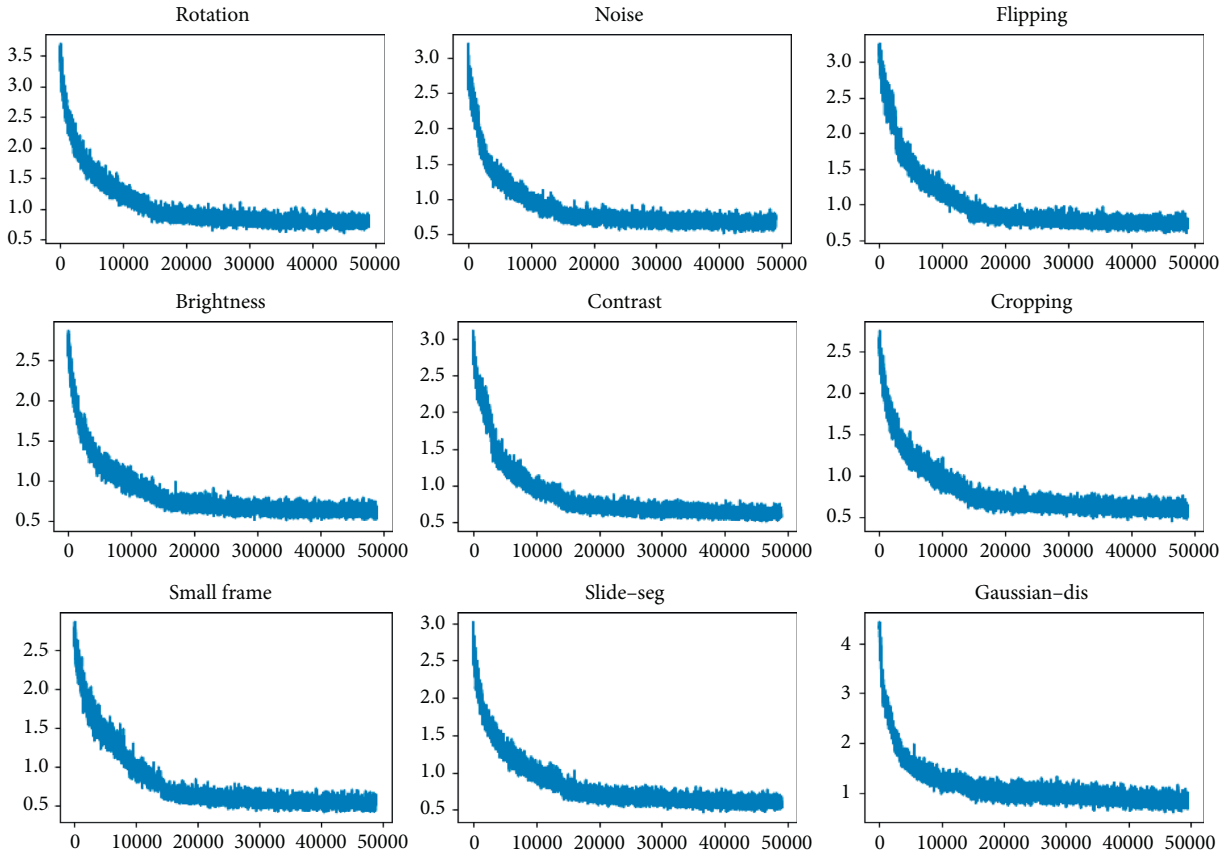

Figure 6: The loss changes of all methods in the training process on YOLO. We can see that the loss value of all methods will eventually converge to less than 1.

It can be observed that no matter what kind of augmentation method is used, there is a step-by-step loss drop between the 10000 and 20000 rounds of iterations, which proves that our learning rate decay strategy is effective under these methods. We have counted the best test results of each model in the above experiments, and these data are helpful for us to judge which methods are helpful for the test of vehicle logo data. The results are as in Table 2.

The experimental results show that the cropping operation improves the results the most in the traditional method, and the $F1$ value increases by 2.63 percentage points. Compared with the original dataset, the $F1$ values of the three methods proposed in this paper increased by 4.11, 4.04, and 5.56 percentage points, respectively. The results show that our method is more effective than the traditional method to help the network to detect the vehicle logo.

We choose the best three of the traditional methods and our proposed method for further experiments to observe the effect of our proposed method in combination. In order to ensure the accuracy of the conclusion, we test the data in the framework of YOLO and Faster R-CNN, respectively, and get the results of $F1$ value in Figure 7.

It can be seen that, compared with the traditional methods, our methods can improve the performance better under the framework of YOLO and Faster R-CNN. We select the best test results of various methods and make them into Table 3.

Our method mainly optimizes the scale diversity and location diversity of the target. Based on the characteristics of large differences between classes and small differences within classes, we propose a new data optimization method. Compared with the traditional method, we mainly focus on two points with the least amount of information in the small dataset of vehicle logo. Experiments show that our idea is effective.

From the experimental results, we can know that the improvement effect of our method on the YOLO framework is better than that of Faster R-CNN. Our method is 3.77 percentage points higher than the traditional method in the

Table 2: The best test results of all methods on YOLO.

	Rotate-on	Noise	Flipping	Brightness	Contrast	Cropping	Small frame	Slide-seg	Gaussian-dis
P	0.8539	0.8521	0.8492	0.8422	0.8641	0.8577	0.9139	0.8862	**0.9151**
R	0.6077	0.6096	0.6170	0.6395	0.6311	0.6433	0.6358	0.6489	**0.6564**
F	0.7101	0.7107	0.7147	0.7270	0.7294	0.7351	0.7499	0.7492	**0.7644**

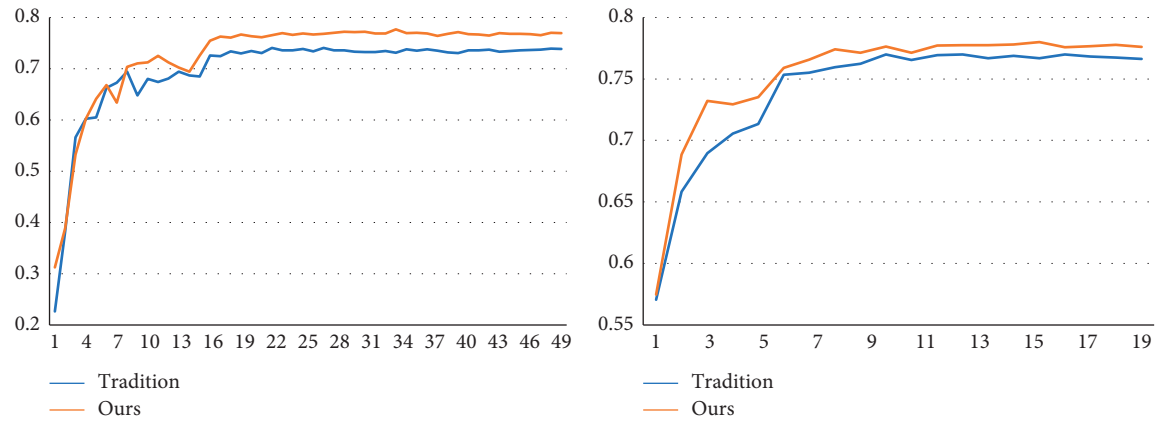

Figure 7: Results of controlled experiments. Although the traditional method sometimes works better than our method in the early stage of training, the final result is that our method is dominant. (a) Result curves of YOLO. (b) Result curves of Faster R-CNN.

Table 3: The best detection results of two networks in each dataset.

	YOLO			Faster R-CNN		
	Origin	Tradition	Ours	Origin	Tradition	Ours
P	0.8466	0.8625	**0.9295**	0.8885	0.9127	**0.9358**
R	0.6096	0.6461	**0.6667**	0.4775	0.6657	**0.6685**
*F*1	0.7088	0.7388	**0.7765**	0.6212	0.7699	**0.7799**

Table 4: Some promotion results show.

	YOLO	Faster R-CNN
Origin		
Tradition		
Ours		

framework of YOLO, which is better than Faster R-CNN. The reason is that we mainly optimize the size and location of the vehicle logo, which is more targeted for the one-stage method like YOLO. Due to the existence of RPN network, Faster R-CNN is not very sensitive to the location of the target, and $F1$ only increases by 1.0 percentage points. As a one-stage method, the YOLO framework can directly return the location of the target from the image. Previous studies have pointed out that CNN is more sensitive to the location of the target in the image, so the regression of the target location in the YOLO framework is more dependent on the dataset. Our method is mainly optimized for the diversification of logo location, so our method performs better under the framework of YOLO. Therefore, although the method proposed in this paper can improve the performance of both networks, it is more obvious to improve the performance of YOLO.

4.3. Results Show. We show some detection effects after data enhancement in Table 4.

In Table 4, the first column of each group of images shows the help of our proposed method for improving the network recall rate. For the image in the first column, the network trained by the original dataset and the dataset enhanced by the traditional method cannot detect the vehicle logo in the image, and the data enhancement method proposed by us can successfully detect the vehicle logo. The second column of each group of images shows the improvement of accuracy of our method. For the error detection in the network trained by the original dataset and the dataset enhanced by the traditional data method, our method can suppress it to a certain extent.

5. Conclusion

In this paper, we propose a series of data optimization methods for small sample vehicle logo dataset. According to the characteristics of small sample vehicle logo dataset, we enhance the dataset from two aspects: size and location. The experimental results show that the methods proposed in this paper can improve the recall and precision compared with the traditional method, whether alone or in combination. For different frameworks, our method is more suitable for the one-stage methods that directly find the target position from the image, so the improvement effect of Faster R-CNN is weaker than that of YOLO.

Data Availability

The image data used to support the findings of this study have been deposited in the Baidu Netdisk repository (https://pan.baidu.com/s/10LG6vZGK_tEV6sGbDx4uug).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant nos. 61972097, 61502105, and 61672158, the Technology Guidance Project of Fujian Province under Grant no. 2017H0015, the Natural Science Foundation of Fujian Province under Grant no. 2018J1798, the Fujian Natural Science Funds for Distinguished Young Scholar under Grant no.2015J06014, the University Production Project of Fujian Province under Grant no. 2017H6008, the Fujian Collaborative Innovation Center for Big Data Application in Governments, and the Fujian Engineering Research Center of Big Data Analysis and Processing.

References

[1] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[2] B. Chen, L. Shi, and Ke. Xiao, "A robust moving object detection in multi-scenario big data for video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 982–995, 2018.

[3] T. Lai, H. Wang, Y. Yan, T.-J. Chin, and W.-L. Zhao, "Motion segmentation via a sparsity constraint," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 973–983, 2017.

[4] Y. Niu, W. Lin, and X. Ke, "CF-based optimisation for saliency detection," *IET Computer Vision*, vol. 12, no. 4, pp. 365–376, 2018.

[5] X. Ke and W. Guo, "Multi-scale salient region and relevant visual keywords based model for automatic image annotation," *Multimedia Tools and Applications*, vol. 75, no. 20, pp. 12477–12498, 2016.

[6] Ke Xiao, L. Shi, W. Guo, and D. Chen, "Multi-Dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2157–2170, 2019.

[7] İ. Türkyılmaz and K. Kaçan, "License plate recognition system using artificial neural networks," *ETRI Journal*, vol. 39, no. 2, pp. 163–172, 2016.

[8] Y. Yuan, W. Zou, Z. Yong et al., "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.

[9] S. Yu, S. Zheng, H. Yang et al., "Vehicle logo recognition based on bag-of-words," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 353–358, IEEE, Krakow, Poland, August 2013.

[10] B. Cyganek and M. Woźniak, "An improved vehicle logo recognition using a classifier ensemble based on pattern tensor representation and decomposition," *New Generation Computing*, vol. 33, no. 4, pp. 389–408, 2015.

[11] D. F. Llorca, R. Arroyo, and M. A. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*, pp. 2229–2234, IEEE, Hague, Netherlands, October 2013.

[12] H. Peng, X. Wang, H. Wang et al., "Recognition of low-resolution logos in vehicle images based on statistical random

sparse distribution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 681–691, 2015.

[13] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015.

[14] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Vehicle logo recognition using whitening transformation and deep learning," *Signal, Image and Video Processing*, vol. 13, no. 1, pp. 111–119, 2019.

[15] A. P. Psyllos, C. N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a SIFT-based enhanced matching scheme," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, 2010.

[16] Y. Liu and S. Li, "A vehicle-logo location approach based on edge detection and projection," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, pp. 165–168, IEEE, Beijing, China, July 2011.

[17] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition," *Iet Intelligent Transport Systems*, vol. 12, no. 8, pp. 939–946, 2018.

[18] J. Redmon, S. Divvala, R. Girshick et al., "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, IEEE, Las Vegas, NV, USA, June 2016.

[19] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society,Honolulu, HI, USA, July 2017.

[20] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[21] R. Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Columbus, OH, USA, June 2014.

[22] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Santiago, Chile, December 2015.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "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.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[26] A. Neubeck and L. V. Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 850–855, IEEE, Hong Kong, China, August 2006.

[27] H. Li and C. Y. Suen, "Robust face recognition based on dynamic rank representation," *Pattern Recognition*, vol. 60, no. 12, pp. 13–24, 2016.

[28] S. K. Biswas and P. Milanfar, "Linear support tensor machine with LSK channels: pedestrian detection in thermal infrared images," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4229–4242, 2017.

[29] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[30] X. Ke, M. Zhou, Y. Niu, and W. Guo, "Data equilibrium based automatic image annotation by fusing deep model and semantic propagation," *Pattern Recognition*, vol. 71, pp. 60–77, 2017.

[31] R. Liu and D. F. Gillies, "Overfitting in linear feature extraction for classification of high-dimensional image data," *Pattern Recognition*, vol. 53, pp. 73–86, 2016.