Journal of Computers Vol. 31 No. 3, 2020, pp. 303-318
doi:10.3966/199115992020063103024

Improved Vehicle Front Target Detection Algorithm Based on Faster R-CNN

Zhi Tan[1], Shuai Tan[1*], Yu Zhu[1]

[1] Beijing University of Civil Engineering and Architecture, Beijing, China
tanzhi@bucea.edu.cn; tanshuai940620@163.com; yuhuohuo0503@163.com

Received 23 March 2020; Revised 1 April 2020; Accepted 23 April 2020

Abstract. Aiming at the problem of low vehicle detection rate in street scene, a forward vehicle target detection algorithm based on improved Faster R-CNN is proposed. Firstly, the deep linear convolutional neural network is used to fully extract the target features; then, for the difficult to detect vehicle targets, the difficulty of mining is introduced in the regional recommendation network to make the training more adequate, and the clustering algorithm is used to determine the length and width ratio of the recommendation frame; In the small target detection problem, the RoI normalization algorithm of bilinear interpolation is introduced into the pooling layer of the target area. Finally, an adaptive optimization algorithm is selected to optimize the parameters. The experiments were performed on the KITTI dataset for training and testing. The results show that the average accuracy of the improved model is 77.1%, 7.39% higher than the original Faster R-CNN, the total training time is reduced by 2.95h, and the total test time is reduced by 9.724s. In this case, the training and detection time is reduced, and the requirements for improving the detection performance of the vehicle ahead can be met.

Keywords: faster R-CNN, vehicle target detection, region proposal network

1 Introduction

Due to the rapid increase in the number of cars in recent years, traffic accidents have also gradually increased, resulting in increasing traffic pressure. Intelligent vehicle target detection has become a hot research direction.

Vehicle detection belongs to the scope of target detection. At the same time, target detection is also a vital part of intelligent traffic management systems and a key research topic of deep learning [1]. With the rapid development of computer technology and the widespread application of computer vision, the use of deep learning image processing technology for target detection has achieved great success, and its related technologies have also continuously made new breakthroughs. In recent years, the computing power of computers has increased rapidly. Hinton [2] and others first proposed the concept of deep learning, which has given deep learning a new starting point in the academic and industrial circles. Since 2012, convolutional neural networks have been applied to major machine vision tasks. Alex Krizhevsky and others first proposed the deep convolutional neural network AlexNet [3], which was 15.3% higher than the second place in large-scale visual recognition challenges. Won the championship with accuracy scores, making the convolutional neural network a core algorithm in target detection.

Target detection is divided into traditional target detection and deep learning-based target detection. Traditional target detection methods are based on support vector machines [4] and directional gradient histograms [5]. The main steps are: (1) generating target suggestion frames; (2) extracting features in each frame; (3) classifier the design of. Traditional target detection methods have the disadvantages of poor portability, inaccurate feature extraction, high complexity and slow speed. Deep learning-based target detection integrates the three steps of traditional detection into the same deep network model, which improves the shortcomings of traditional target detection. At the same time, because the

* Corresponding Author

convolutional neural network extracts multiple features through training, breaking the traditional target detection algorithm requires manual work. Limitations of structural features.

The R-CNN proposed by Girshick R [6] et al. Used the deep model to extract image features for the first time, and used the AlexNet network as the network structure for feature extraction. Good results are obtained, but there is a problem of double counting. With the continuous development of R-CNN, Fast R-CNN [7] solves the problem of redundant computing and adds a simplified Spatial Pyramid Pooling [8] (SPP), so that the training and testing processes can be combined in Together, the speed of training and testing is obviously accelerated, but the speed still cannot meet the real-time requirements. Faster R-CNN is the representative algorithm of the current two-stage target detection algorithm. On the basis of Fast R-CNN, a region proposal network (RPN), shared convolutional layer and feature map are added to effectively shorten the candidate box Extract time to achieve training and testing of rapid target detection. However, it still takes a lot of time to use SS for feature extraction. The representative algorithm of the single-stage target detection algorithm is YOLO, and its main idea is to solve the regression problem, using an independent and complete network to judge the output of the results, making the YOLO algorithm have the advantages of fast reading and low false detection rate, but the YOLO algorithm For short-range targets and small-scale targets, the detection effect is not good. To compensate for the shortcomings of the YOLO algorithm, the SSD algorithm is proposed. This algorithm combines the RPN network structure and trains a new model, which effectively improves the detection of the target detection algorithm Precision. The RCNN series has a high detection accuracy rate for object detection, but the detection speed is not enough. The YOLO series algorithm has some deficiencies in the convenience of detection accuracy, but the detection speed is faster.

Vehicle detection not only needs to ensure real-time performance but also ensure its accuracy, but due to the complex scenes, many interference factors seriously affect the detection efficiency. Faster R-CNN can automatically extract the feature information of the target in the training sample, which has certain scale and displacement. Invariance, there is relatively high detection accuracy and detection speed in target detection, but the model has poor detection effect on problems such as severe occlusion of vehicles and pedestrian targets, small scales of distant targets, and different light intensities, so the original Faster R -The CNN model has not yet fully met the requirements for vehicle detection. In order to better solve the above problems that affect the accuracy of vehicle target detection, based on deep learning methods, the classic Faster R-CNN network model used in vehicle detection scenarios is improved. The improved Faster R-CNN network can achieve more accuracy And rapid identification, and solve the problem of low detection rate caused by interference factors in vehicle target detection.

2 Faster R-CNN Framework Analysis

2.1 Overall Framework

The Faster R-CNN network consists of two parts: the RPN network and the Fast R-CNN network. The RPN network is used to extract candidate regions that may contain targets, called the region of interest (RoIs); the Fast R-CNN network is used to The purpose of classification is to distinguish the target from the background, and to refine the target area box. Faster R-CNN integrates feature extraction, candidate frame extraction, border regression, and classification in a network, which greatly improves the speed of object detection. The network structure is shown in Fig. 1.

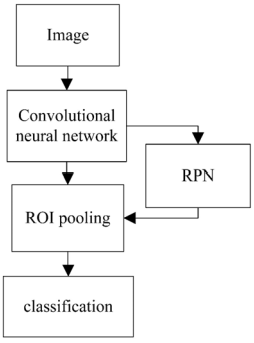


Fig. 1. Faster R-CNN network structure

The implementation process of vehicle target detection method based on Faster R-CNN is shown in Fig. 2. First, you need to convert the data set to VOC 2007 format for model input. The processed data set is input to the feature extraction layer, and the feature map is obtained after the convolution layer and the pooling layer. In the RPN network, first use a 3 × 3 sliding window to perform a similar convolution operation. Each pixel of each feature map corresponds to the original image as an area, as shown in Fig. 2 to Fig. 3. And generate anchors and K different recommended area frames, then convert the high-dimensional information into low-dimensional information, input it to the fully connected layer, and finally divide the foreground and background of the recommended area frame generated by the RPN network and generate the recommended area frame After the processing is completed, the feature map generated by the feature extraction layer is mapped to the region recommendation frame generated by the RPN network, processed into a feature map of the same size through the ROIs network, and then entered into the fully connected layer, and finally classified and the target frame regression In the training process of the network, the feature extraction layer is trained with parameters pre-trained in Image Net, and after a certain number of iterations, the model performance is optimized.

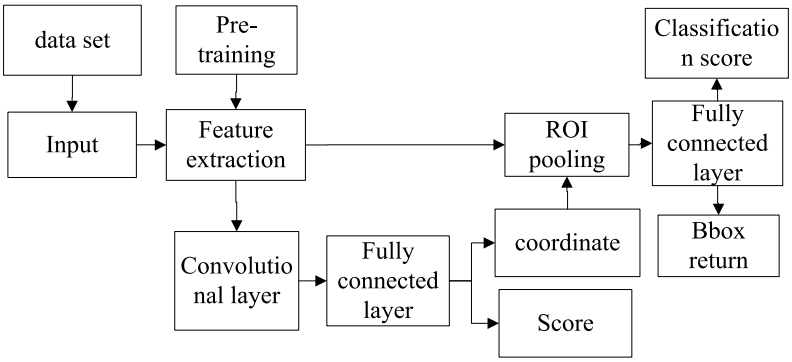


Fig. 2. Faster R-CNN algorithm implementation flowchart

During training, alternately train the RPN network and Fast R-CNN, the steps are as follows:
Step 1: RPN network initialization and training.
Step 2: Fast R-CNN network initialization and training.
Step 3: Train the RPN network twice.
Step 4: Train the Fast R-CNN network twice.
Step 5: Jointly train the Fast R-CNN network and the RPN network.
During the training of the Faster R-CNN network, the loss function is divided into two parts: classification loss and regression box loss, that is,

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum L_{reg}(t_i, t_i^*). \quad \textbf{(1)}$$

Where $\frac{1}{N_{cls}}\sum L_{cls}(p_i, p_i^*)$ represents classification loss, p_i represents the probability that the candidate area prediction result is a vehicle, When the p_i^* value is 1, it indicates that the area is a positive sample, When the p_i^* value is 0, it means that the area is a negative sample, $L_{cls}(p_i, p_i^*)$ represents the log loss of the two categories and can be described as

$$L_{cls}(p_i, p_i^*) = -\log[p_i p_i^* + (1-p_i^*)(1-p_i)]. \quad \textbf{(2)}$$

In Equation 1, $\lambda \frac{1}{N_{reg}}\sum L_{reg}(t_i, t_i^*)$ represents regression box loss, $t_i = \{t_x, t_y, t_w, t_h\}$ represents the predicted offset within the region, t_i^* represents the actual offset.

$$L_{reg}(t_i, t_i^*) = \sum_{i\in\{x,y,w,h\}} smooth_{L1}(t_i, t_i^*). \quad \textbf{(3)}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, |x| \le 1 \\ |x| = 0.5, \text{other} \end{cases}. \quad \textbf{(4)}$$

2.2 RPN Network Works

The RPN network is used to extract vehicle target candidate areas. It is a full convolutional network. It shares the convolution layer with the feature extraction part, and uses the unique network to slide the feature map output by the feature extraction layer to obtain the target candidate area with the target score. The input of the RPN network is the d n × n space regions of the feature map, which are mapped to d-dimensional vectors. Finally, the low-dimensional vectors are input into two fully connected layers, namely the candidate region classification layer and the candidate region frame regression layer. The window position center corresponds to the original image to predict k target region suggestion frames, so for each position, the candidate region classification layer has 2k outputs, that is, the probability of belonging to the foreground and background, and the candidate region frame regression layer has 4k outputs, that is, pan and zoom Parameters, as shown in Fig. 3.

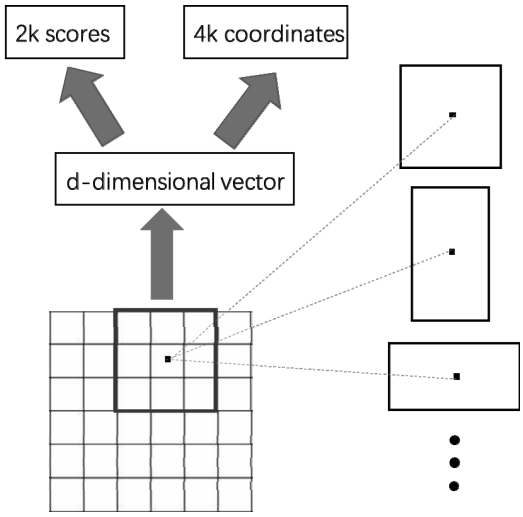


Fig. 3. RPN structure diagram

3 Faster R-CNN Framework Analysis

3.1 Problems with the Current Algorithm

Faster R-CNN is the representative algorithm of the current two-stage target detection. Its advantages are high model accuracy, that is, high positioning and detection rate, and its network structure backbone network uses convolutional neural networks, which can share the calculation amount and reduce Complexity, real-time monitoring can be basically achieved if GPU is used in network implementation, but the disadvantage is that the training and detection time of the network model is long, so that real-time detection cannot be achieved, and the vehicle and pedestrian targets existing in the vehicle detection task are severely blocked, Distant targets with small scales and different light intensities cannot meet the accuracy required by the detection task.

In order to make the target detection algorithm better suitable for vehicle target detection in street scenes, the improvement based on the Faster R-CNN model is mainly divided into deep linear convolutional neural network feature extraction, vehicle target candidate area generation based on difficult example mining, RoI normalization based on bilinear interpolation, hyperparameter determination based on clustering algorithm and parameter optimization algorithm selection.

3.2 Deep linear Convolutional Neural Network Feature Extraction

Faster R-CNN commonly used convolutional neural network structures are ZF network [9] and VGG-16 network [10]. VGG is a convolutional neural network architecture for image classification. It uses a linear structure and passes different numbers of convolutional layers. The pooling layer and the pooling layer are superimposed on each other to form a linear convolutional neural network. The data is processed in turn from the first layer until the last layer is completed. The convolution layer operation is similar to the filtering process. A sliding window with a convolution kernel performs convolution on a local area of the input picture and generates a feature map. The convolution operation formula can be expressed as

$$f(x,y) * w(x,y) = \sum_{x=-a}^{a} \sum_{x=-b}^{b} w(s,t) f(x-s, y-t). \tag{5}$$

Assuming k different convolution kernels in the convolutional neural network [11], k different feature maps are extracted, and the feature values at the k-th feature map (i, j) of the first layer are expressed as

$$z_{i,j,k}^{l} = w_k^l x_{i,j}^l + b_k^l. \tag{6}$$

Where w_k^l and b_k^l represent the weight and offset of the k-th convolution kernel in a layer. $x_{i,j}^l$ represents the input matrix of a layer.

The pooling layer is a discretization process that acts on samples. It downsamples feature maps to reduce dimensions. The feature map is divided into several blocks, and the maximum value in each block is taken to reduce the redundant information in the features and enhance the robustness of the model.

The deep network helps extract more feature information [12]. For the vehicle target detection task, based on the linear structure of VGG, the convolution layer and pooling layer are divided into five blocks, named Block1 ~ Block5 in turn. The structure of the deep linear convolutional neural network is shown in Fig. 4.

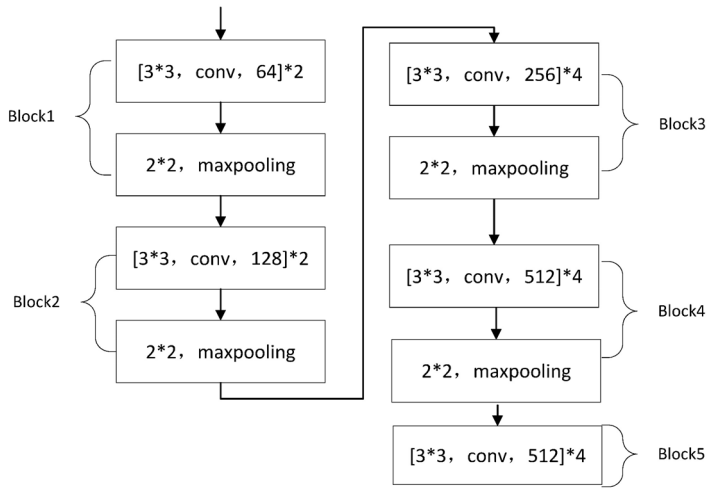


Fig. 4. Schematic diagram of deep linear convolutional neural network

The convolutional layers all use 3 × 3 small-scale convolution kernels with a step size of 1. The number of convolutional layers is 2, 2, 4, 4, 4, respectively. The number of convolutional channels can be doubled to effectively use computing resources, so Set the number of Block1 convolution channels to 64, Block2 to 128, Block3 to 256, and Block4 and Block5 to 512. The pooling layer uses a 2 × 2 sliding window with a step size of 2. With the non-linear activation function RELU, the size of the feature map does not change. The visualization of each block feature map is shown in Fig. 5.

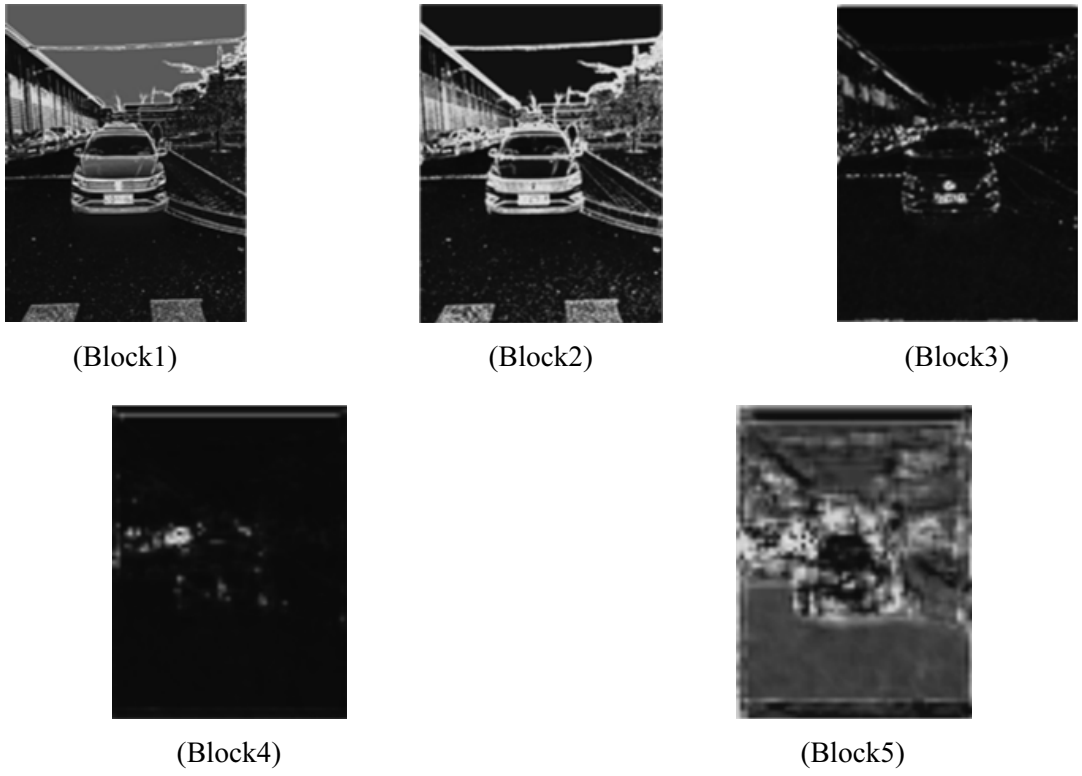


Fig. 5. Feature map visualization

In order to compare the performance of ZF network, VGG-16 network and deep convolutional neural network structure more intuitively [13], three network structures were used as feature extraction layers, and Faster R-CNN was trained using KITTI dataset. A total of four stages were performed. 240000 iterations. Draw a loss curve and record the accuracy of the model. The loss curve is shown in Fig. 6.

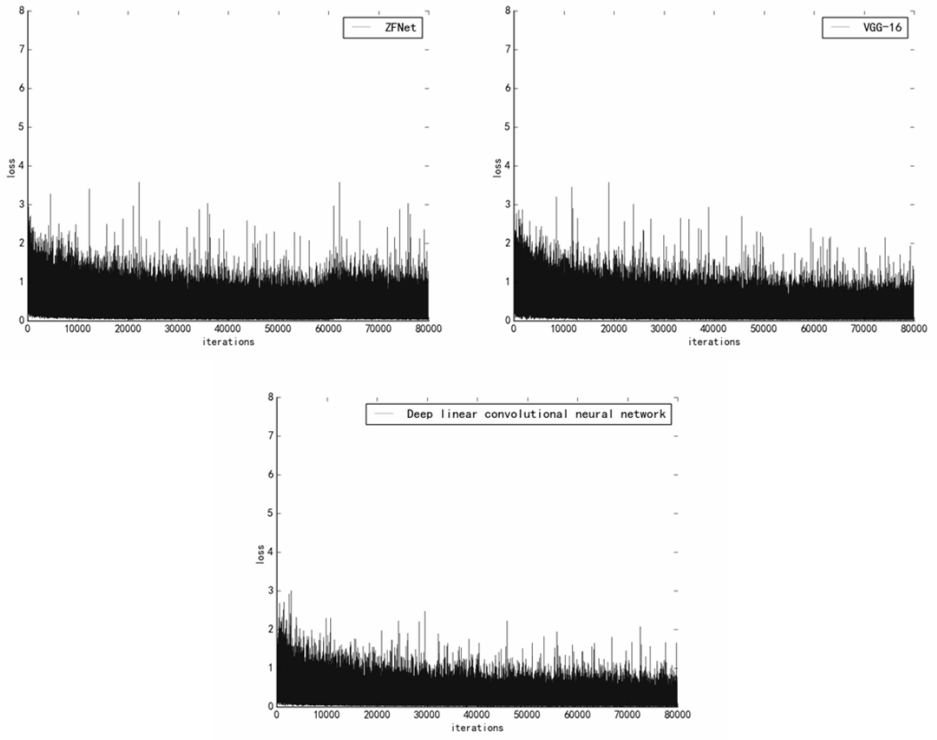


Fig. 6. Loss curves of three feature extraction layers

It can be seen from Fig. 6 that although the Faster R-CNN algorithm using the ZF network structure and the VGG-16 network structure can converge to a certain range, the results are not satisfactory, and the designed deep linear convolutional neural network structure is used to improve the curve oscillation. Large amplitude problems, and the curve is stable in a small amplitude range.

3.3 Vehicle Target Candidate Region Generation Scheme Based on Difficult Case Mining

The regional recommendation network is used to generate regional candidate image blocks on the feature map. The RPN network is the core part of the Faster R-CNN network. Its input is the feature map generated by the feature extraction part. The 3 × 3 convolution kernel is used to perform convolution on the feature map to obtain a three-dimensional matrix data structure. One-dimensional vector, each pixel of the feature map finally becomes an anchor. Relative to the input image, each Anchor predicts three sizes and three aspect ratios for a total of nine scales, which are input to two fully connected layers for classification and rectangular box regression, respectively.

The main purpose of the RPN network is to generate an accurate recommended area frame for the detection target, and calculate the Intersection over Union (IOU) of the generated candidate area and the label area. The calculation method is shown in Fig. 7. The numerator is the intersection area of the candidate area and the target area. And the denominator is the area where the candidate region and the target region merge.

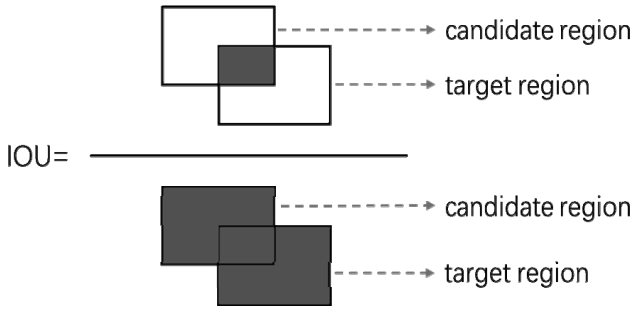


Fig. 7. IOU calculation method

Settings with an overlap rate greater than 0.7 are positive samples, otherwise they are negative samples. Therefore, during the training of the RPN network, the number of negative samples generated is much larger than the positive samples, and most of the negative samples are easily classified samples. The original RPN network used the cross-entropy function to calculate the loss. The cross-entropy of each training sample was directly summed, and the positive and negative samples had the same weight, so that the negative sample loss accounted for most of the total loss. The loss function at this time was optimized during the training iteration process. Slow and not optimal, i.e.

$$L = -y\log y' - (1-y)\log(1-y') = \begin{cases} -\log y' & y=1 \\ -\log(1-y') & y=0 \end{cases}. \tag{7}$$

In view of this problem, the difficult case mining idea [14] was introduced into the model and described as

$$L_{fl} = \begin{cases} -\alpha(1-y')^{\gamma}\log y' & y=1 \\ -(1-\alpha)y'^{\gamma}\log(1-y') & y=0 \end{cases}. \tag{8}$$

Focus loss [15] increases the modulation factor based on the hyper-parameters of the cross-entropy γ With shared weights α, By setting γ to adjust the weight of difficult and easy samples, By setting α to control the shared weight of the positive and negative samples to the total loss, the problem of uneven proportion of the balanced positive and negative samples itself is solved, Among them, the value of γ is 2 and the value of α is 0.25.

This method can realize the difficulty of vehicle target detection and solve the problem of uneven positive and negative samples in the training data set.

3.4 RoI Layer Normalization Based on Bilinear Interpolation

The original Faster R-CNN network uses the RoI Pooling operation. The main steps are divided into two steps: (1) quantizing the integer point coordinate values of the candidate box boundaries; (2) dividing the quantized boundary area into k × k units on average. The boundary of each unit is quantified, and the back propagation formula of RoI Pooling is expressed as

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i*(r,j)] \frac{\partial L}{\partial y_{rj}}. \tag{9}$$

x_i represents the pixels on the feature map before pooling; y_{rj} represents the j-th point of the r-th candidate region after pooling. After two quantizations, the candidate frame has a certain deviation from the original regression position, that is, a mismatch problem, resulting in low detection accuracy. The improved model introduces the RoI layer normalization method RoI Align using bilinear interpolation. The main steps are divided into three steps:

(1) Iterate through each candidate area, keeping the floating point number boundary without quantization;

(2) Divide the candidate area into k × k units, and the boundary of each unit is not quantized;

(3) Calculate the fixed four coordinate positions in each unit, and calculate the values of the four positions by the method of bilinear interpolation. Assuming that the values of the four points are Q11 = (x1, y1), Q12 = (x1, y2), Q21 (x2, y1) and Q22 = (x2, y2), first perform linear interpolation in the x direction, that is

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x_2 - x_1}{x_2 - x_1} f(Q_{21}), R_1 = (x, y_1). \tag{10}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x_2 - x_1}{x_2 - x_1} f(Q_{22}), R_2 = (x, y_2). \tag{11}$$

Then perform linear interpolation in the y direction

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2). \tag{12}$$

Get the final result as

$$\begin{aligned} f(x,y) \approx & \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y_2 - y). \\ & + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y - y_1). \end{aligned} \tag{13}$$

Finally, the maximum pooling operation is performed, and the back propagation formula of RoI Align is expressed as

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [d(i, i*(r,j) < 1](1 - \Delta h)(1 \Delta w) \frac{\partial L}{\partial y_{rj}}. \tag{14}$$

$d(\,.\,)$ represents the distance between two pixels. This algorithm solves the problem of region mismatch caused by two quantizations in RoI Pooling operation, and has better performance for small target detection.

3.5 Determination of Hyperparameters Based on Clustering Algorithm

Faster R-CNN presets three types of aspect ratios for the area recommendation frame, namely 0.5, 1, and 2. This aspect ratio is suitable for most target detection recommendation frames. It is not targeted for vehicle target detection in street scenes. Although Faster R-CNN adjusts the position and scale of

recommended candidate frames, when the candidate frame generated by the network When the data sets have large scale differences, even the iterative adjustment cannot guarantee that the coincidence between the candidate frame and the target true labeling frame reaches a predetermined IOU threshold, thereby affecting the accuracy of target detection. In order to make the generated recommendation area frame more suitable for street view, the KITTI data set is used to perform dimensional clustering analysis using the K-means algorithm, trying to find the optimal solution, and using the average intersection ratio to evaluate the performance of the solution. (Including the length and width of all target labeling boxes in the training set), using d = 1-IOU as the distance formula, and output the clustering results and K cluster center points. The K values are assigned from 1 respectively. The K-means clustering results for different K values are shown in Fig. 8. Different gray levels represent different categories, and the cluster center points of each category are represented by five-pointed stars.

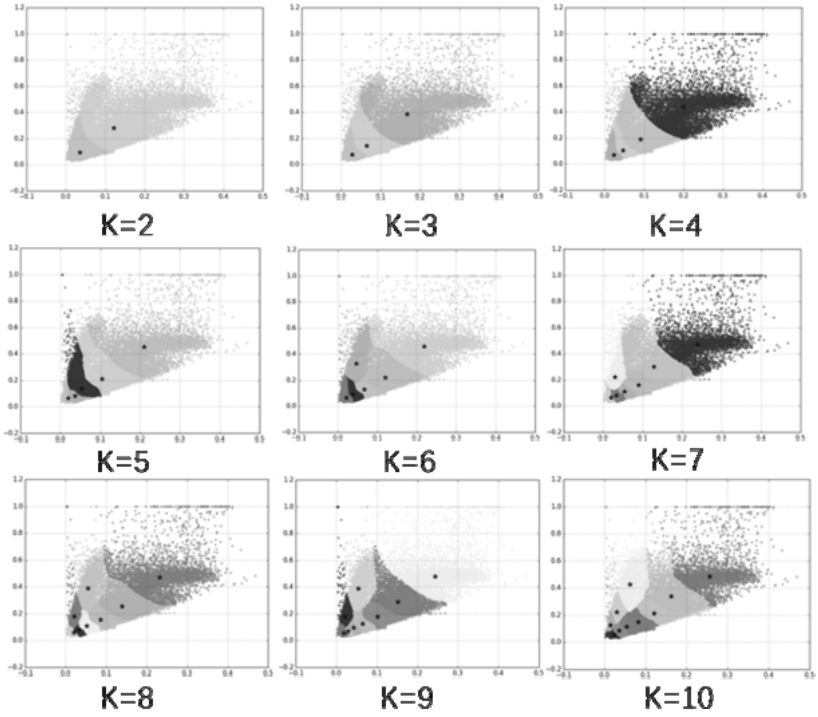


Fig. 8. K-means clustering results based on different K values

Observe the change of the average crossover ratio. When the value of K is greater than 4, the change of the average crossover ratio becomes smaller and smaller, as shown in Fig. 9.

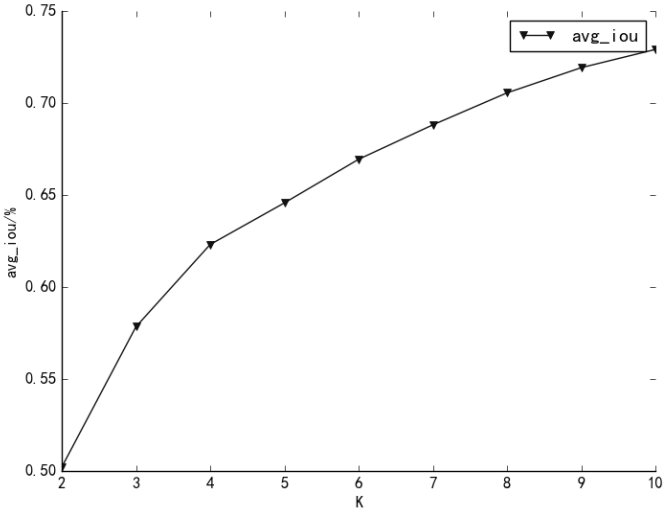


Fig. 9. Results of average cross-comparison based on different K values

The inflection point value K = 4 is selected as the optimal clustering result, and the length-to-width ratio of the optimal target labeling frame obtained by multiple experiments is 0.13, 0.34, 0.44, 0.46, as shown in Fig. 10. According to the disparity in the size of the vehicle pedestrian labeling frame in the KITTI dataset, three scales are set, which are 128, 256, and 512, a total of 12 recommended area frames with different aspect ratios and scales, which is an increase from the 9 scales of the original model Got 3 kinds. These recommendation boxes of different sizes can effectively capture large-scale vehicle targets and dense small-scale pedestrian targets.

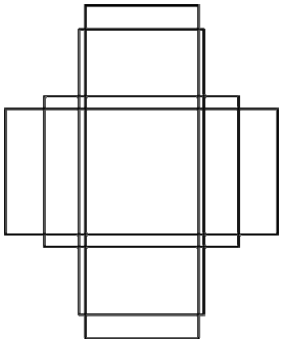


Fig. 10. Schematic diagram of the aspect ratio of the four target callout boxes

3.6 Selection of Parameter Optimization Algorithm

The parameter optimization algorithm used in the original Faster R-CNN network is the Momentum algorithm. The Momentum algorithm uses the Stochastic Gradient Descent (SGD) method of momentum. With the concept of momentum in physics, when updating parameters, the current parameters are The change will be affected by the last parameter change, similar to inertia in physics, making the parameter update more stable and the learning speed faster. Momentum algorithm is expressed as

$$\begin{cases} v_t = av_{t-1} + \eta \Delta J(W_t, X^{(is)}, Y^{(is)}) \\ W_{t+1} = W_t - v_t \end{cases}. \tag{15}$$

v_t is the speed accumulated at t, a represents the magnitude of the momentum, generally 0.9 (represents 10 times the maximum speed for SGD), W_{t+1} is the model parameter at time t.

Adam algorithm name comes from Adaptive Moment Estimation. Compared with the SGD algorithm, Adam's optimization algorithm is fast and not easy to fall into a local optimum. The algorithm makes full use of the first-order moment mean and second-order moment mean, and dynamically adjusts the parameter learning rate so that each iteration has a corresponding accurate range. At the same time, the parameters are updated more smoothly. The Adam algorithm updates the parameter expression as

$$\theta \leftarrow \theta - \frac{\eta}{\sqrt{v_t + \varepsilon}} \cdot m_t . \tag{16}$$

3.7 Network Parameter Modification Strategy

When using deep learning methods for target detection tasks, the most important step is to improve network performance by modifying network parameters, so the parameters of Faster R-CNN network are modified from the following aspects:

(1) The main function of the learning rate is to control whether the objective function can converge to the minimum value after a large number of iterations and how much time it takes to reach the minimum value during the training process. Choosing the right learning rate can make the performance of the model reach the highest performance in the shortest possible time. Therefore, the learning rate is an important hyperparameter of the deep learning network model. When the learning rate is too large, the model learning speed is fast, but it is easy to appear that the calculated loss value is very large, and the shock is obvious. When the learning rate is set too small The network learning speed is slow, and the

problem of overfitting is prone to occur, and the speed of loss value convergence is relatively slow. In the network model of this paper, 0.001 is selected as the initial learning rate, and the adjustment parameter that controls the learning rate is set to 0.1 The learning rate can slowly decrease with the increase of training times.

(2) Batch Size represents the number of samples required for each training of the network. Its size is not only closely related to the optimization degree and speed of the network, but also affects the amount of GPU memory occupied by training. Before the concept of Batch Size was proposed, in order to make the direction of gradient descent more accurate during model training, the entire training set was read into each iteration for calculation, resulting in a large difference in the final gradient value, and a unified learning rate could not be used. When using a batch size suitable for the network, the computational efficiency of memory can be improved through parallelized calculation. In order to achieve a balance between memory size and efficiency, this article sets the Batch Size to 256.

(3) The activation function is a method used by the neural network to add non-linear factors to the network, and its purpose is to fit various curves and enhance the expression ability of the network. The main idea of the Relu function can be understood as taking the maximum value, its left half is always in the state, and the right half is in the activated state, which is in line with the activity mode of human neurons. The function fitting speed is 6 times, which is currently the most widely used nonlinear activation function. In this paper, the network finally selects the Relu function as the nonlinear activation function of the neural network.

4 Experimental Results and Analysis

In order to verify the effectiveness of the improved model, KITTI, the world's largest dataset of autonomous driving scenes, was selected to verify the model performance. Using the data set format of VOC2007, the data set was divided into 6,732 training sets and 748 test sets. Model training was performed using labeled training sets, and the test set tested and evaluated the improved network model.

By improving and adjusting the Faster R-CNN network model, a comparative experiment under different methods is performed on the vehicle detection data set. The open source framework TensorFlow is used for experimental research. The experimental environment uses Window 10 operating system, i7 quad-core CPU, clocked at 2.8Ghz, 16G memory, and uses Anaconda, pycharm and python3 software platforms.

For the vehicle target detection experiment [14], the test results are divided into four types: TP, FP, TN, and FN. The specific meanings are shown in Table 1.

Table 1. Meaning of classification results

Real situation	Positive prediction	Negative prediction
Positive example	TP	FN
Negative example	FP	TN

In order to verify the detection quality of the experiment, the following results are used to evaluate the indicators:

Average accuracy (AP). First set a set of thresholds. If the recall is greater than each threshold, a corresponding accuracy will be obtained. AP is the average of these accuracy rates. The higher the value, the more accurate it is. The higher the detection accuracy of the algorithm.

Mean AP Precision (MAP). In a multi-class classification task, the model will identify multiple classes. MAP is the average of all known APs. The higher the value, the higher the algorithm's detection accuracy.

Precision and Recall (P-R) curve, where the horizontal axis represents the recall rate, the vertical axis represents the precision rate, the P-R curve is plotted as this, and the recall rate formula is expressed as

$$Recall = TP/(TP + FN) \tag{17}$$

The precision formula is expressed as

$$Precision = TP/(TP + FP) \tag{18}$$

4.1 Comparison of the Results of the Three Feature Extraction Layers

The commonly used feature extraction layers of Faster R-CNN are ZFNet and VGG-16. In order to verify the effectiveness of the proposed deep linear convolutional neural network, three feature extraction layers are put into Faster R-CNN network without changing the rest In the case of comparative tests, the experimental results are shown in Table 2. It can be seen from Table 2 that the model MAP using the proposed deep linear convolutional neural network is higher than the ZF network structure and VGG-16 network structure by 5.12% and 0.43%, and the car AP value is increased by 1.39% and 0.42%, respectively. The linear convolutional neural network can effectively enhance the feature extraction capability of the model, thereby improving the detection accuracy of the model.

Table 2. Comparison of accuracy of different feature extraction models

Algorithm	MAP/%	AP-car/%
VGG-16	68.63	79.9
ZFNet	63.94	78.93
Deep linear convolutional neural network	69.71	80.17

4.2 Comparison of Four Parameter Optimization Algorithm Results

Because the data set with a very large amount of data is used to train the network, the training speed of the network is very slow, so selecting the appropriate parameter optimization algorithm can quickly improve the detection efficiency of the target detection model. In order to achieve the best network target detection effect, Momentum algorithm, NAG algorithm, AdaGrad algorithm and Adam algorithm are used to compare the performance of training vehicle images. During the training process, starting from 30,000 iterations, the accuracy of the model using the Adam optimizer is higher than the accuracy of the model using the remaining three parameter optimizers. The line graph of the trend of accuracy during the iteration process is shown in Fig. 11. The final detection accuracy results of the four optimizers are shown in Table 3.

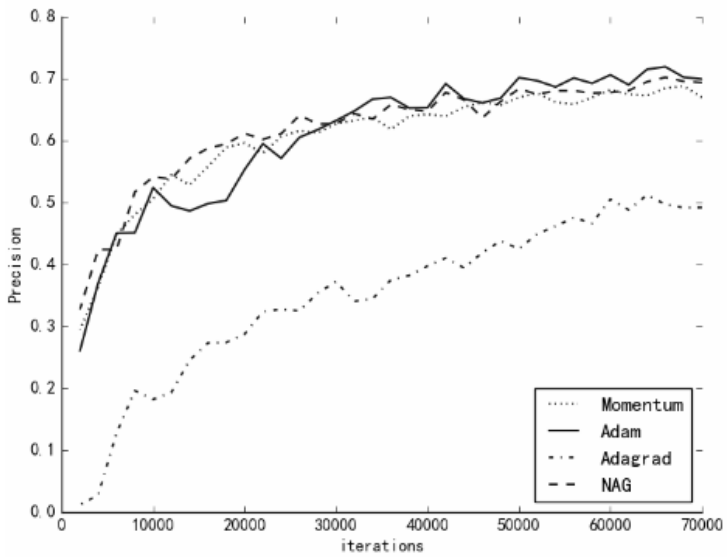


Fig. 11. Performance comparison of four optimization algorithms

Table 3. Comparison of the accuracy rates of the four optimizers

Optimizer	MAP/%	AP-car/%
Momentum	70.55	80.62
NAG	70.97	80.24
AdaGrad	51.2	75.07
Adam	73.44	81.14

It can be seen from Table 3 that the Faster R-CNN detection MAP value using the Adam adaptive algorithm is 2.89%, 2.47%, and 22.24% higher than the Momentum optimizer, NAG optimizer, and AdaGrad optimizer, respectively. In order to improve the accuracy of model detection, the adaptive optimization algorithm Adam with the best effect is finally selected as the improved model parameter optimization algorithm.

4.3 Performance Test with Different Improvement Strategies

In order to verify the improvement effect of the proposed method in depth, a comparison experiment of different improvement strategies was carried out. The purpose was to test the effectiveness of the different improvement strategies in the proposed method. Each subsequent experiment was based on the improvement strategy of the previous experiment, and was sequentially stacked and compared The experiments all set the same learning rate of 0.01. The experimental comparison results are shown in Table 4, where $\surd$ represents use and $\times$ represents unused.

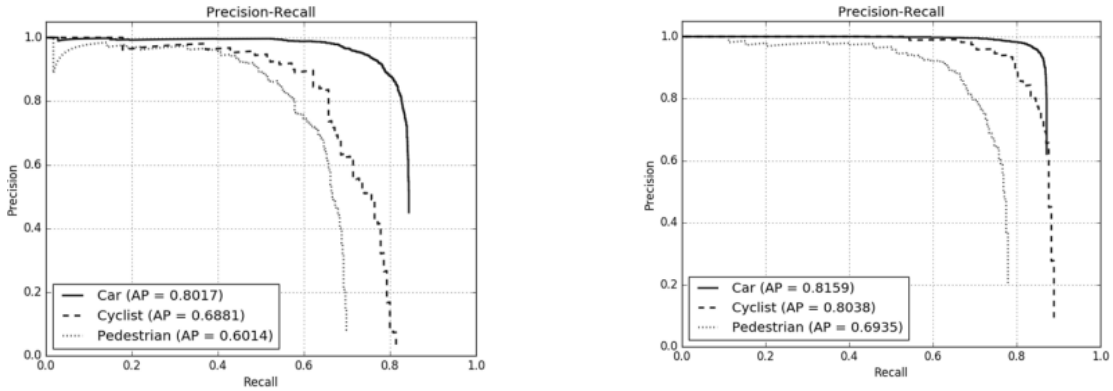
Table 4. Comparison of the effect of different methods on model promotion

Number	Anchor number	Deep linear convolutional neural network	Adaptive optimization algorithm	Difficult case mining	Cluster Anchor aspect ratio	Bilinearly interpolated ROI	AP-Car/%	MAP/%
1	9	×	×	×	×	×	79.9	68.63
2	9	√	×	×	×	×	80.32	69.06
3	9	√	√	×	×	×	81.07	73.44
4	9	√	√	√	×	×	81.13	74.11
5	12	√	√	√	√	×	81.15	75.26
6	12	√	√	√	√	√	81.59	77.1

It can be seen from Table 4 that different improvement strategies have different contributions to the network. Before the improvement, the model's detection accuracy has steadily increased by using the superposition of different improvement strategies. The original Faster R-CNN model has a MAP of 69.71%. Strategies have different effects on model promotion. Among them, the Adam adaptive optimizer algorithm has the most obvious effect on the model effect. After superimposing five improvement strategies into the original model in sequence, the MAP of the final model detection reached 77.1%, which was 7.39% higher than the original model.

4.4 Comparison of P-R Diagrams of Models Before and After Improvement

The P-R curve is an important indicator for judging the performance of the model. The comparison between the original model and the improved model P-R curve is shown in Fig. 12. It can be seen from the figure that the P-R curve of the improved model is smoother and the balance point is better than the original model.



(a) Original model (b) Improve the model

Fig. 12. P-R curve comparison chart

4.5 Time Performance Analysis

In order to compare the impact of different improvement strategies on model training and detection time performance, the average single iteration time of the original model training, the total training time when the model performance reached the optimal, the average detection time, and the total time after testing 748 images were used as evaluations Standard, the time performance results of the model training part are shown in Table 5, and the time performance results of the test part are shown in Table 6.

Table 5. Time performance analysis of training part

Algorithm	Total training times	Average single iteration time /s	Total training time /h
Improve the model	72000	0.937	18.74
original Faster R-CNN	84000	0.928	21.65

Table 6. Time performance analysis of training part

Algorithm	Total test set	Average detection time /s	Total detection time /s
Improve the model	748	0.173	129.404
original Faster R-CNN	748	0.186	139.128

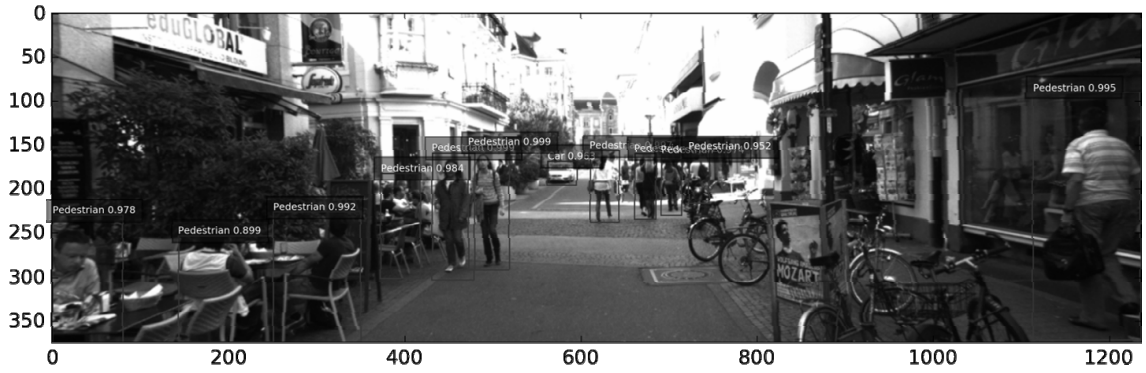
As can be seen from the above table, the original Faster R-CNN model takes an average of 0.186s to take each picture, the average single iteration time during the training process is 0.928s, and the total training time of the model is reduced by 2.95h after the training is improved The total test time is reduced by 9.724s. Overall, the improvement strategy improves the accuracy of vehicle detection in street scenes while reducing the total training time, proving the effectiveness of the improvement program.

4.6 Detection Effect Icon

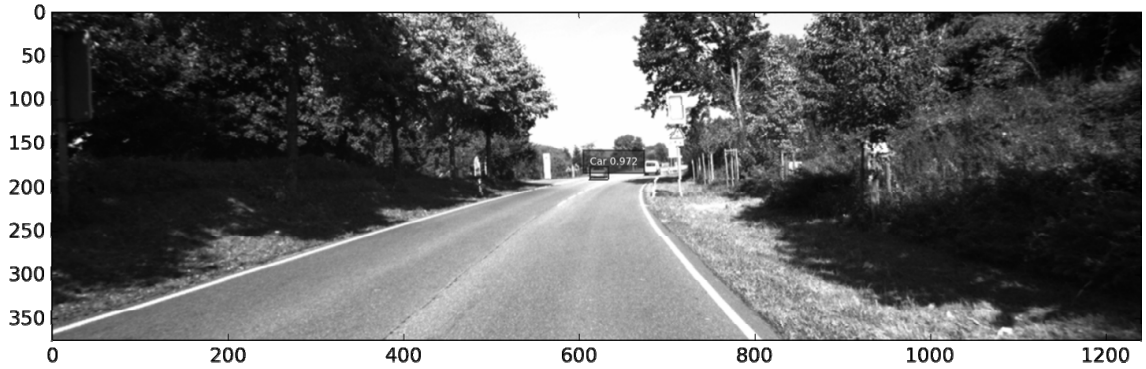
The improved model was used to detect images randomly selected from the KITTI data set, including different postures, partial occlusions, severe overlaps, different light intensities and too small targets. The detection results are shown in Fig. 13. It can be seen from Fig. 13(a) that the improved model can detect pedestrian targets with different postures. From Fig. 13(b), it can be seen that the improved model can adapt well to large light intensity and small scale In the case of target detection, it can be seen from Fig. 13(c) and Fig. 13(d) that the improved model is also robust to different degrees of occlusion.

5 Conclusion

Aiming at the problems of severe occlusion of vehicle and pedestrian targets in street scenes, small distant target scales, and low detection accuracy caused by different light intensities, a forward vehicle target detection algorithm based on Faster R-CNN is proposed. The main improvement strategy of this algorithm is to design deep linear convolution Neural network feature extraction, vehicle target candidate area generation based on difficult case mining, RoI normalization based on bilinear interpolation, hyperparameter determination based on clustering algorithm, and selection of parameter optimization algorithms. Experimental results show that the improved algorithm has better detection performance than the original network. In the future work, a large number of small target detection problems under complex backgrounds will be studied to achieve higher detection accuracy.

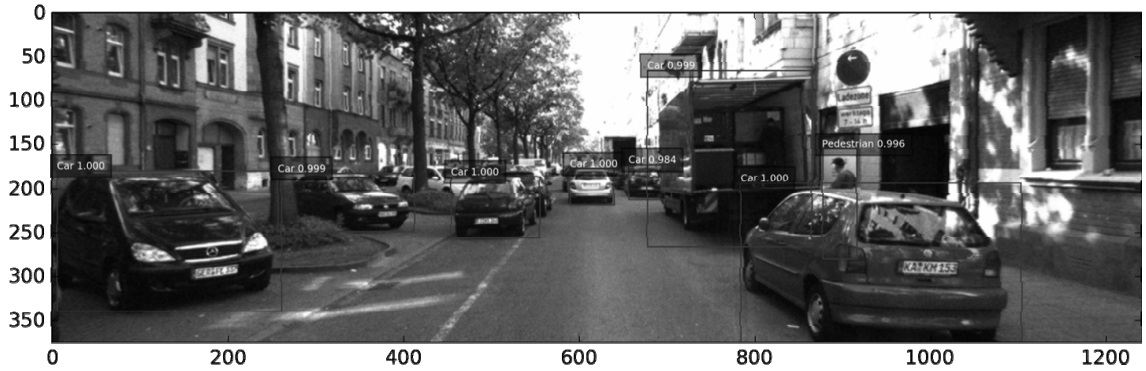


(a) Pedestrian detection icon in different poses



(b) Light intensity and small vehicle detection icon

(c) Vehicle blocking pedestrian detection icon



(d) Pedestrian severe overlap occlusion detection icon

Fig. 13. Schematic diagram of improved Faster R-CNN detection effect

References

[1] X.-Y. Zhang, H.-B. Gao, J.-H. Zhao, M. Zhou, Summary of autonomous driving technology based on deep learning, Journal of Tsinghua University (Natural Science Edition) 58(04)(2018) 438-444.

[2] G.E. Hinton, Reducing the dimensionality of data with neural networks, Science 313(5786)(2006) 504-507.

[3] A. Krizhevsky, I Sutskeever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Communications of the ACM 60(2)(2012) 1097-1105.

[4] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2(2)(1998) 121-167.

[5] N. Daral, Histograms of Oriented Gradients for Human Detection, CVPR 177(05)(2005) 886-893.

[6] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR. 80(1)(2014) 580-587.

[7] R, Girshick, Fast R-CNN, Computer Science 169(4)(2015) 1440-1448.

[8] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Transactions on Pattern Analysis & Machine Intelligence 37(9)(2014) 346-361.

[9] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, ECCV. 53(1)(2014) 818-833.

[10] K.Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science 1556(9)(2014) 680-694.

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR 90(1)(2016) 770-778.

[12] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, Computer Science 28(5)(2014) 413-425.

[13] X.-F. Wu, J.-X. Zhang, X.-C. Xu, Gesture recognition algorithm based on Faster R-CNN, Journal of Computer Aided Design and Graphics 41(3)(2018) 468-476.

[14] J.-W. Li, C.-W. Qu, S.-J. Peng, Y. Jiang, SAR image ship target detection based on generative confrontation network and online hard case mining, Journal of Electronics and Information Technology 41(1)(2019) 143-149.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis & Machine Intelligence PP(99)(2017) 2999-3007.