Preceding Vehicle Detection Based on Optimized Faster R-CNN Algorithm

Yongqing Wang¹,a, Guochen Cui²,b, Shufeng Wang¹,*, Junyou Zhang¹,c

¹ College of Transportation, Shandong University of Science and Technology, Qingdao, China
² Jinan Bresee Technology Co., Ltd., Jinan, China

a wangyongqing93@163.com, b c949443083@163.com, c junyouzhang@sdust.edu.cn
*Corresponding author e-mail: shufengwang@sdust.edu.cn

Abstract. Preceding vehicle detection is still a challenge for unmanned driving technology. Deep learning has achieved great success in target detection. Among them, the Faster R-CNN algorithm is a classic representative. However, the algorithm still has some room for improvement in detection accuracy. By analyzing the problems of Faster R-CNN in the detection of occluded vehicles, taking the target detection post-processing algorithm Soft-NMS as the research object, two new penalty coefficients, inverse proportional penalty coefficient and exponential penalty coefficient, were proposed. It further improves the algorithm's detection accuracy of the blocked vehicle in front.

Keywords: Faster R-CNN algorithm; preceding vehicle detection; occlusion; Soft-NMS algorithm; penalty coefficient.

1. Introduction

Compared with human driving, unmanned vehicles have outstanding environmental awareness and path planning capabilities, which can greatly reduce the traffic accident rate and ease the pressure of traffic congestion [1]. In the unmanned driving system, perception is an extremely important part. Real-time detection of the preceding vehicle can provide sufficient prior conditions for subsequent decision-making and planning, thereby avoiding traffic accidents [2]. In recent years, preceding vehicle detection has become a research hotspot due to its important role in autonomous vehicles.

At present, the methods for detecting vehicles in front are mainly divided into traditional machine learning methods and deep learning methods based on neural networks. Traditional machine learning methods mainly extract vehicle features through feature extraction operators such as HOG (histogram of oriented gradient) [3] and Haar-like [4], and then input these features into classifiers such as SVM (support vector machine) [5] and AdaBoost [6] to complete vehicle detection. These methods have the disadvantages of high time complexity, insufficient model learning, and inability to adapt to the requirements of feature diversity. At the same time, it cannot be applied to large samples and cannot accurately include the current complex road traffic environment of unmanned driving [2]. At the same time, machine learning methods are difficult to solve the problems of occlusion and false detection. In this context, it is more effective to use deep learning methods to detect vehicles in front. With the continuous development of deep learning, the convolutional neural network can effectively extract high-
dimensional features of images due to its parameter sharing, local connection, and down-sampling structure of the simulation vision processing method, which greatly improves the detection accuracy. Therefore, it is widely used in vehicle target detection tasks, and the effect far exceeds traditional algorithms. Faster R-CNN (faster region with convolutional neural network) [7] is a target detection framework based on region recommendations and convolutional neural networks proposed by Microsoft Research in 2015 that can perform end-to-end learning. Zhang [8] built a fast and accurate road target detection algorithm (FAROD) based on Faster R-CNN, and improved the detection performance of small targets by introducing a deconvolution structure. Frameworks such as SSD (single shot multibox detector) [9] and YOLO (you look only once) v2 [10], which are known for fast detection, also draw on many ideas of Faster R-CNN. As a classic algorithm for target detection, it is used by many scholars in the field of vehicle detection.

In the process of vehicle detection, the detection of the car in front of the occlusion is a difficult problem to be solved urgently. When the front vehicle is blocked by other vehicles or obstacles, it is easy to miss detection or low detection accuracy, so that the target vehicle cannot be accurately identified. The Faster R-CNN algorithm mainly uses the NMS (Non-maximum suppression) algorithm to remove the redundant target detection frame, while the optimized NMS algorithm can improve the detection accuracy of the vehicle in front of the occlusion. Bodla [11] proposed a Soft-NMS algorithm with penalty coefficient. This method does not need to retrain the original model, and can be easily integrated into any target detection algorithm using NMS, reducing the rate of missed detection. Zhao [12] applied the Soft-NMS algorithm to the task of detecting targets such as vehicles. Compared with the NMS algorithm, the Soft-NMS algorithm can improve the accuracy of the PASCAL VOC 2007 data set by 1% and 2%, so using the Soft-NMS algorithm can improve the detection accuracy of the car ahead. Soft-NMS algorithm proposes linear weighted and Gaussian weighted penalty coefficients. However, no scholars consider applying other types of penalty coefficients to the Soft-NMS algorithm to explore its impact on the detection accuracy.

Therefore, aiming at the detection accuracy of the car in front of the occlusion, this paper introduces the Soft-NMS algorithm into the Faster R-CNN algorithm. The Soft-NMS algorithm is optimized and two new penalty coefficients are introduced to improve the detection accuracy of the occluded vehicle. First of all, the paper introduces the overall structure and design principle of Faster R-CNN, RPN (the region proposal network) and NMS. Then the working principle of Soft-NMS algorithm is introduced and two new penalty coefficients are introduced to optimize Soft-NMS algorithm. Finally, the new penalty coefficients are verified by experiments, and the detection accuracy of Faster R-CNN algorithm is further improved.

2. Faster R-CNN algorithm

2.1. How Faster R-CNN works

As shown in Fig. 1, the work of Faster R-CNN is divided into four parts. The first part is image feature extraction. The image is processed by a convolutional neural network (VGG16 as an example) to obtain a feature map. The second part is RPN. According to the feature map input through the front layer network, the network will determine the candidate region and the general location of the target in the image, including the detection frame and whether it is the foreground and background. The third part is RoiPooling. Through the proposed region and feature map information, RoiPooling maps the proposed region to the feature map. The fourth part is classification and regression. Through the feature map of the proposed region, the cls layer and the reg layer determine the position and category of the target in the image, respectively.
RPN is a fully convolutional neural network, and its workflow is shown in Fig. 2. The input of this network is the feature map generated by the previous convolutional layer. First, use the sliding window of $n \times n$ ($n = 3$ in this article) to traverse the feature map, generate a new 512-dimensional feature map, and generate $k$ anchors. Then, the 512-dimensional feature is mapped on the low-latitude vector through a $1 \times 1$ convolution operation. These vectors will be used in the cls layer and reg layer.

The picture will be adjusted to a fixed size before entering the network. Through a series of convolution operations (taking the VGG16 network as an example), a feature map with a size of $50 \times 38$ is finally generated. Because each feature point corresponds to $k$ anchors, a total of $50 \times 38 \times k$ anchors are mapped on the original image. Through the processing of the cls layer and the reg layer, each anchor will obtain four parameters including two scores of the target and the corresponding position. According to these parameters, after a post-processing process, about 300 suggested areas can finally be generated. The post-processing process is shown in Fig. 3.
Faster R-CNN will generate detection bounding boxes and scores for specific categories of targets. Adjacent detection bounding boxes often have related scores, which will increase the false positives of the test results. In order to avoid this situation, the Faster R-CNN algorithm applies the NMS algorithm to remove redundant detection frames. The working principle of the NMS algorithm is as follows.

First, the algorithm will generate a series of detection frames $B_i$ and a series of confidence scores $S_i$ ($i=1, 2, ..., j, ...$). Then select the detection frame $B_j$ with the highest confidence score and its confidence score $S_j$. Then determine the $IoU$ value of the detection frame and other detection frames $B_i$ ($i \neq j$). Subsequently, the confidence score of $B_i$ ($i \neq j$) is updated according to Equation (1). If the confidence score of $B_i$ ($i \neq j$) is 0, the detection frame will be removed. Then select the remaining detection frames except $B_j$ and repeat the above operation until all target detection frames are selected.

$$S_i = \begin{cases} S_j, & IoU < Threshold \\
0, & IoU \geq Threshold \end{cases}$$  \hspace{1cm} (1)

Among them, in the design of Faster R-CNN algorithm, $Threshold = 0.3$.

2.2. Soft-NMS algorithm

Although the NMS algorithm can effectively reduce the false positives of the detection results, the method is a greedy algorithm. It forces the removal of the detection bounding box adjacent to the highest confidence score. If a real target appears in the overlapped area and the overlapped area is too large, the target will be deleted by mistake, causing the target to be unrecognizable and reducing detection accuracy. Therefore, the Soft-NMS algorithm came into being based on the above reasons. The core idea of the Soft-NMS algorithm is to no longer violently delete all target detection frames with $IoU$ greater than the threshold, but to reduce the confidence score of the detection frame. Compared with Equation (1), Soft-NMS smooths it and proposes a penalty coefficient $\lambda$ based on linear weighting and Gaussian weighting. The above two penalty coefficients are determined by Equation (2) and Equation (3) respectively, and the corresponding confidence score is determined by Equation (4). This soft-NMS algorithm based on penalty coefficients effectively improves the detection effect of Faster R-CNN algorithm on occluded targets by reducing the confidence score instead of directly deleting the target detection frame.

$$\lambda = \begin{cases} 1, & IoU(B_i, B_j) < Threshold \\
1-IoU(B_i, B_j), & IoU(B_i, B_j) \geq Threshold \end{cases}$$  \hspace{1cm} (2)

$$\lambda = e^{-\frac{IoU(B_i, B_j)}{\delta}}, \quad i \neq j$$  \hspace{1cm} (3)

$$S_i = \lambda S_j$$  \hspace{1cm} (4)

In the Equation, $IoU(B_i, B_j)$ represents the intersection ratio between the detection frame $B_i$ to be processed and the detection frame $B_j$ with the highest built-in reliability in the same detection frame set, $S_i$ is the confidence score of the detection frame $B_i$, $Threshold = 0.3$, $\delta = 0.3$.

3. Faster R-CNN optimizes the detection accuracy of the vehicle in front of the occlusion

Based on the above research, this paper proposes a further optimization of the Soft-NMS algorithm. First, the influence of the threshold on the penalty coefficient is discussed. Then, two other types of penalty coefficients are introduced according to the curve of the penalty intensity of the penalty coefficient changing with $IoU$, so as to improve the detection effect of the Faster R-CNN algorithm on the vehicle in front of the occlusion.
### 3.1. The impact of the threshold on the original penalty coefficient

Threshold is also called critical value, refers to the lowest or highest value that an effect can produce. The Soft-NMS algorithm is not easy to set the threshold. If the threshold is too small, the target will be missed, and if the threshold is too large, the target will be falsely detected. Therefore, by adjusting different thresholds, the impact of maintaining the penalty intensity of the penalty coefficient within the threshold on the detection accuracy of the preceding vehicle is explored.

Regarding the linear penalty coefficient, this section explores the detection results of the Faster R-CNN algorithm when the penalty coefficient value is 1 when the threshold is 0.3, 0.2, 0.1, and 0, respectively. The test results are shown in TABLE 1, where AP means average precision.

| Threshold | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0   |
|-----------|-----|-----|-----|-----|-----|-----|
| AP value (%) | 79.07 | 79.96 | 80.32 | 80.29 | 80.24 | 79.33 |

It can be seen from TABLE 1 that with the decrease of the threshold, the detection accuracy of the algorithm for the vehicle in front of the occlusion also gradually decreases. The following conclusions can be drawn from the above results:

1. The reduction of the threshold means that the penalty coefficient can maintain the same penalty intensity in a smaller range. The decrease in detection accuracy means the decrease in the ability to maintain the penalty intensity accompanied by the decrease in the threshold. The detection effect changes in a negative direction.

2. When the threshold is 0, the detection accuracy of the algorithm for the blocked vehicle in front is close to 1% with the threshold. This means that maintaining a penalty intensity of 1 within a certain threshold has a positive effect on the detection effect.

Because the original Gaussian penalty coefficient does not set a threshold, the threshold is set in the Gaussian penalty coefficient and the penalty coefficient is 1 within the threshold range to prove the above point. Regarding the Gaussian penalty coefficient, this paper also explores the detection results of keeping the penalty coefficient value of 1 when the threshold is 0.3, 0.2, 0.1, and 0, respectively. The test results are shown in TABLE 2.

| Threshold | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0   |
|-----------|-----|-----|-----|-----|-----|-----|
| AP value (%) | 79.46 | 80.42 | 80.98 | 80.96 | 80.91 | 80.24 |

Through the analysis of the experimental results in TABLE II, the following conclusions can be drawn:

1. Compared with the original Gaussian penalty coefficient, by introducing the threshold, the penalty intensity can be maintained within a certain threshold range, which has a positive effect on the improvement of detection accuracy.

2. Similar to the linear penalty coefficient, the reduction of the threshold in the Gaussian penalty coefficient brings about the weakening of the ability to maintain the penalty intensity, which has a certain negative impact on the detection effect.

Through the analysis of the Gaussian penalty coefficient and the linear penalty coefficient, it can be found that keeping the penalty coefficient value of 1 within a certain threshold range will have a positive impact on detection. This not only validates the view that the optimized Gaussian penalty coefficients are not as effective as linear penalty coefficients, but also provides ideas for the design of new penalty coefficients.
3.2. Analysis and optimization of penalty coefficient

Cui [13] proposed an optimized Soft-NMS algorithm, which multiplies the linear penalty coefficient and the Gaussian penalty for many times. The new Equation for calculating the penalty coefficient $\lambda^Q$ is shown in the Equations (5) and (6).

$$
\lambda^Q = \begin{cases} 
1, & \text{if } \text{IoU}(B_i, B_j) < \text{Threshold} \\
(1 - \text{IoU}(B_i, B_j))^\sigma, & \text{if } \text{IoU}(B_i, B_j) \geq \text{Threshold}
\end{cases}
$$

(5)

$$
\lambda^Q = \left( \frac{-\text{IoU}(B_i, B_j)}{\delta} \right)^\sigma, \quad i \neq j
$$

(6)

Under this process of multiple multiplication, the variation curve of the penalty intensity of the penalty coefficient with $\text{IoU}$ is shown in Fig. 4.

![Fig. 4 The relationship between penalty intensity, IoU and Q](image)

(a) Optimized linear penalty coefficient  (b) Optimized Gaussian penalty coefficient

In this optimization algorithm, the processing of multiplying the penalty coefficient $Q$ times enables the algorithm to improve the detection accuracy of the vehicle in front of the block by 1%-2%. At the same time, it is found that compared with the linear penalty coefficient, the Gaussian penalty coefficient has a weak ability to maintain its own penalty intensity, which is the reason why the detection accuracy of the Gaussian penalty coefficient is lower than the linear penalty coefficient.

Based on the above research, this paper proposed an inverse proportional penalty coefficient and an exponential penalty coefficient according to the curve change law of the penalty function coefficient.

1) Inverse proportional penalty coefficient

It is known that the linear penalty coefficient and the Gaussian penalty coefficient have been multiplied multiple times to achieve a better detection effect of occluded vehicles. From Fig. 4, it is found that the curves of the two penalty coefficients are gradually concave when multiplied multiple times, which is similar to the curve of the inverse proportional function $y = k/x \quad (x > 0)$. Therefore, the curve of the penalty coefficient $\lambda$ is an inverse proportional function in the interval $(N_t, 1)$.

Similar to the linear penalty coefficient, the penalty intensity is maintained within a certain threshold range, and the penalty coefficient value is maintained at 1. Therefore, the penalty coefficient should have the same penalty intensity as the linear penalty coefficient when $\text{IoU}(B_i, B_j)$ is between 0 and $N_t$. Therefore, the design function of the inverse proportional penalty coefficient $\lambda$ is shown in Equation (7).

$$
\lambda = \begin{cases} 
1, & \text{if } \text{IoU}(B_i, B_j) < \text{Threshold} \\
\sigma \left( \frac{-\text{IoU}(B_i, B_j)}{\delta} \right)^\sigma, & \text{if } \text{IoU}(B_i, B_j) \geq \text{Threshold}
\end{cases}
$$

(7)

In the Equation, $\sigma = 0.1$.

In the previous analysis, it is known that the detection effect of the optimized linear weighted penalty coefficient is better than the Gaussian weighted penalty coefficient. Therefore, the designed inverse proportional penalty coefficient $\lambda$ is compared with the optimized linear weighted penalty coefficient. When the threshold $N_t$ is 0.3, the function curve of the inverse proportional penalty coefficient is shown in Fig. 5(a), and the comparison after adding this function curve to the optimized linear weighted penalty...
coefficient curve is shown in Fig. 5(b). Through comparison, it is found that the penalty coefficient $\lambda$ still has some characteristics of the optimized linear weighted penalty coefficient.

2) Exponential penalty coefficient

Similar to the Gaussian penalty coefficient, the exponential function also has the characteristic that the larger the $\text{IoU}$ value, the stronger the penalty intensity. The exponential penalty coefficient $\lambda$ based on the exponential function is as follows:

$$\lambda = a^{\text{IoU}(B_i, B)}$$  \hspace{1cm} (8)

In the Equation, $a$ is a constant.

This article chooses to set three a values, respectively 0.1, 0.05 and 0.01. The schematic diagram of the exponential penalty coefficient under different values of $a$ is shown in Fig. 6(a).

It can be seen from Fig. 6(a) that, compared with the optimized Gaussian penalty coefficient, the exponential penalty coefficient has a similar penalty curve, and the penalty intensity is different under different values of $a$.

It is known that for the penalty coefficient, maintaining a certain penalty intensity within the threshold range has a positive effect on the detection effect, so the exponential penalty coefficient with a threshold range is shown in Equation (9):

$$\lambda = \begin{cases} 1 & \text{IoU}(B_i, B) < \text{Threshold} \\ a^{\text{IoU}(B_i, B)} & \text{IoU}(B_i, B) \geq \text{Threshold} \end{cases} \hspace{1cm} (9)$$

In this case, the curve of the exponential penalty coefficient is shown in Fig. 6(b). According to the above two designs, the optimized confidence score is shown in Equation (10).

$$S_i = S_i \times \lambda \hspace{1cm} (10)$$
4. Experimental verification
The design experiments of inverse proportional penalty coefficient and exponential penalty coefficient are verified. Firstly, the evaluation index and training environment are introduced. Then, the effects of the two newly introduced penalty coefficients on the preceding vehicle detection accuracy of Faster R-CNN are evaluated respectively. Finally, evaluate the effect of their ability to maintain the penalty intensity within the threshold range on the detection effect.

4.1. Evaluation indicators and environmental configuration
The most commonly used model evaluation indicators are $P$ (Precision), $R$ (Recall), and $mAP$ (mean Average Precision). The $P$ represents the model's ability to identify related targets, and is the percentage of correct predictions. $R$ refers to the target's ability to find related targets, and is the percentage of all targets predicted to be correct. For two classification problems, the $AP$ (Average Precision) represents the performance of the classifier, that is, the area of the $P-R$ curve, to reflect the evaluation effect of the balance between precision and recall. Among them, the $P$ and $R$ are determined by Equation (11) and Equation (12) respectively:

$$P = \frac{TP}{TP + FP} \quad (11)$$
$$R = \frac{TP}{TP + FN} \quad (12)$$

Among them, the meanings of TP, FP and FN are shown in TABL 3.

Tab.3 Interpretation table of positive and negative examples

| Real result | Forecast result |
|-------------|-----------------|
| **Positive** | **Negative** |
| Positive    | True positive (TP) | False negative (FN) |
| Negative    | False positive (FP) | True negative (TN) |

The training environment is configured as shown in TABLE 4.

Tab.4 Environment configuration for training

| Requirements               | Parameter                  |
|----------------------------|----------------------------|
| Operating system           | Ubuntu 16.04              |
| Deep learning framework    | Caffe                      |
| Central processing unit (CPU) | Intel Core i7-6700    |
| Graphics processing unit (GPU) | Nvidia Geforce GTX 1060 |
| Training method            | End-to-end                 |
| VGG model                  | VGG_CNN_M_1024             |
| Learning rate              | 0.001                      |
| Number of iterations       | 70,000                     |

4.2. Analysis of experimental results
The dataset used in this paper is the KITTI dataset. The models are trained on the dataset and verified by experiments. First, for the inverse proportional penalty coefficient, this paper selects the threshold by jointly adjusting the parameters. In the process of constantly changing the threshold size, keep other network structures unchanged, and take the final detected $AP$ value as the evaluation criterion. Because of the curve nature of the inverse proportional function, this paper tests the detection effect of the inverse proportional penalty coefficient when the threshold $N_t$ is 0.3, 0.2 and 0.1 respectively. The test results are shown in TABLE 5. At the same time, the detection effect of using the inverse proportional penalty coefficient is further tested. Fig. 7 (a) and (b) respectively show the detection diagrams without applying the inverse proportional penalty coefficient and applying the inverse proportional penalty coefficient.
Tab. 5 Test results of inverse proportional penalty coefficient

| Threshold($N_t$) | 0.3 | 0.2 | 0.1 |
|------------------|-----|-----|-----|
| AP value (%)     | 80.76 | 80.70 | 80.53 |

(a) The inverse proportional penalty coefficient was not applied. (b) The inverse proportional penalty coefficient was applied.

Fig. 7 Detection map of occluded vehicles in front

Through the analysis of the experimental results in TABLE V, it is found that:

1. Compared with the original linear penalty coefficient ($\text{threshold} = 0.3$) and the original Gaussian penalty coefficient ($\text{threshold} = 0$) applied by the Soft-NMS algorithm, the detection accuracy of the inverse proportional penalty coefficient is higher than that of the above two penalty coefficients no matter what threshold the inverse proportional penalty coefficient is. It is proved that the designed inverse proportional penalty coefficient is more effective than the original two penalty coefficients, and the detection effect is better.

2. With the continuous decrease of the threshold, the detection effect of the algorithm for the occluded preceding vehicle is also gradually weakened. This also verifies that the ability of penalty coefficient to maintain the intensity of penalty within the threshold range is weakened, and its effect on the detection effect is negative.

As for the exponential penalty coefficient, this paper verifies the exponential penalty coefficient in the case of $a=0.1$, $a=0.05$ and $a=0.01$ respectively. First of all, this paper verifies the detection effect of the algorithm for the occluded preceding vehicle when the exponential penalty coefficient is applied without considering the threshold as shown in Equation (8). The test results are shown in TABLE 6.

Tab. 6 The test results of exponential penalty coefficient are applied (without considering the threshold)

| Parameters | $a =0.1$ | $a =0.05$ | $a =0.01$ |
|------------|---------|---------|---------|
| AP value (%) | 80.46 | 80.33 | 80.01 |

Then, we continue to test the detection effect of the algorithm on the blocked vehicle in front when the exponential penalty coefficient is applied after considering the threshold ($\text{threshold value} = 0.3$) as shown in Equation (9). The test results are shown in TABLE 7. When the threshold value is considered, the detection diagram of the blocked vehicle in front is shown in Fig. 8.

Tab. 7 The test results of exponential penalty coefficient are applied(considered the threshold)

| Parameters | $a =0.1$ | $a =0.05$ | $a =0.01$ |
|------------|---------|---------|---------|
| AP value (%) | 80.77 | 81.02 | 81.22 |
Fig. 8 Detection of blocked vehicles in front with exponential penalty coefficient (considered the threshold)

Through the analysis of the above experimental results, the following conclusions can be drawn:

1) When the threshold is not considered and the parameter $a$ is 0.1 and 0.05 respectively, the detection effect of exponential penalty coefficient is better than that of linear penalty coefficient and Gaussian penalty coefficient. Although the detection effect is worse when the parameter $a$ is 0.01, it is better than the original NMS algorithm, which proves the effectiveness of the exponential penalty coefficient.

2) When the threshold is 0.3, the detection effect of applying the exponential penalty coefficient is better than the original linear penalty coefficient and Gaussian penalty coefficient under different parameter $a$ value. When $a$ is 0.01, the detection effect is better. It is proved that a new type of penalty coefficient designed to imitate the optimized linear penalty coefficient and Gaussian penalty coefficient curve change trend has a positive effect on the detection effect.

3) When the threshold is not considered, with the continuous decrease of parameter $a$, the worse the detection effect is. When considering the threshold, the detection effect is better with the continuous decrease of $a$. Through the comparison of Fig. 6, it is found that when the threshold is not considered, the ability of the exponential penalty coefficient to maintain its own penalty intensity is worse with the continuous decrease of $a$. After considering the threshold, it can be guaranteed that with the change of parameter $a$, the exponential penalty coefficient maintains the penalty intensity within the threshold range, which is also the reason for the above changes. It also verifies the positive effect of the penalty coefficient within the threshold to maintain the penalty intensity.

To sum up, the inverse proportional penalty coefficient and exponential penalty coefficient proposed in this paper have better detection results than linear penalty coefficient and Gaussian penalty coefficient under certain conditions.

5. Conclusion
In order to improve the detection accuracy of the Faster R-CNN algorithm for the preceding vehicle, this paper makes corresponding optimizations for the detection of occluded vehicles. Aiming at the Soft-NMS algorithm, this paper proposes two new penalty coefficients: inverse proportional penalty coefficient and exponential penalty coefficient. Experiments verify that the two new penalty coefficients have better detection effects than linear penalty coefficients and Gaussian penalty coefficients. At the same time, by further considering the threshold factor, the detection effect has a better performance under certain conditions, which verifies the effectiveness of this algorithm.

Acknowledgments
This work was supported by the Natural Science Foundation of Shandong Province under Grant no. ZR2019MF056.
References

[1] Han X. Deep Learning Based Scene Recognition for Autonomous Driving[D]. Guangzhou: Sun Yat-sen University, 2017.

[2] Qin H. Research on Forward Vehicle Detection Algorithm Based on Fusion of Vision and Millimeter Wave Radar[D]. Hunan University, 2019.

[3] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2005:886-893.

[4] VIOLA P, JONES M J. Robust Real-Time Face Detection[J]. International Journal of Computer Vision, 2004,57(2) :137—154.

[5] Dollar P, Appel R , Belongie S , et al. Fast Feature Pyramids for Object Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(8):1532-1545.

[6] Wen X, Shao L , Xue Y , et al. A rapid learning algorithm for vehicle classification[J]. Information Sciences An International Journal, 2015, 295(C):395-406.

[7] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.

[8] Zhang Q, Wan C, Qin, et al. Road object detection algorithm based on convolutional neural network [J].Computer Engineering and Design,2019,40(07):2052-2058.

[9] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]// European Conference on Computer Vision. Springer, Cham, 2016.

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

[11] Bodla N, Singh B, Chellappa R, et al. Soft-NMS—improving object detection with one line of code[J]. IEEE International Conference on Computer Vision, 2017: 5562–5570.

[12] Zhao W, Yan H, Shao X. Object detection based on improved non-maximum suppression algorithm[J].Journal of Image and Graphics,2018,23(11):1676-1685.

[13] Cui, Wang, Liu, et al. Preceding Vehicle Detection Using Faster R-CNN Based on Speed Classification Random Anchor and Q-Square Penalty Coefficient[J]. 2019.