

Article

# Preceding Vehicle Detection Using Faster R-CNN Based on Speed Classification Random Anchor and Q-Square Penalty Coefficient

Guochen Cui<sup>1</sup>, Shufeng Wang<sup>1,\*</sup>, Yongqing Wang<sup>1</sup>, Zhe Liu<sup>1</sup>, Yadong Yuan<sup>1</sup> and Qiaoqiao Wang<sup>2</sup>

- <sup>1</sup> College of Transportation, Shandong University of Science and Technology, Huang dao District, Qingdao 266590, China; c949443083@163.com (G.C.); wangyongqing93@163.com (Y.W.); lz2019hu@163.com (Z.L.); yyd863319721@163.com (Y.Y.)
- <sup>2</sup> Menaul School Qingdao, NO.17 Wenhai Road, Ocean Silicon Valley, Qingdao 266200, China; wangqiaoqiao145@163.com
- \* Correspondence: shufengwang@sdust.edu.cn; Tel.: +86-186-0532-6013

Received: 23 August 2019; Accepted: 9 September 2019; Published: 12 September 2019



**Abstract:** At present, preceding vehicle detection remains a challenging problem for autonomous vehicle technologies. In recent years, deep learning has been shown to be successful for vehicle detection, such as the faster region with a convolutional neural network (Faster R-CNN). However, when the host vehicle speed increases or there is an occlusion in front, the performance of the Faster R-CNN algorithm usually degrades. To obtain better performance on preceding vehicle detection when the speed of the host vehicle changes, a speed classification random anchor (SCRA) method is proposed. The reasons for degraded detection accuracy when the host vehicle speed increases are analyzed, and the factor of vehicle speed is introduced to redesign the anchors. Redesigned anchors can adapt to changes of the preceding vehicle size rule when the host vehicle speed increases. Furthermore, to achieve better performance on occluded vehicles, a Q-square penalty coefficient (Q-SPC) method is proposed to optimize the Faster R-CNN algorithm. The experimental validation results show that compared with the Faster R-CNN algorithm, the SCRA and Q-SPC methods have certain significance for improving preceding vehicle detection accuracy.

**Keywords:** Faster R-CNN algorithm; preceding vehicle detection; vehicle speed; random anchors; penalty coefficient

# 1. Introduction

The development of an autonomous vehicle system that enhances safety and traffic efficiency is in progress continuously; this system requires a road environment detection facility in order to control the host vehicle [1,2]. The detection of preceding vehicles plays a decisive role in realizing rational planning of the driving path, maintaining the correct distance, and ensuring driving safety for autonomous vehicles [3,4]. Recently, preceding vehicle detection has become a research hotspot due to its necessity for autonomous vehicles, and many detection algorithms have been proposed [5–8].

At present, methods of preceding vehicle detection are mainly divided into the traditional machine learning method and the deep learning method. The traditional machine learning method extracts vehicle features by feature extraction operators such as histogram of oriented gradient (HOG) [9], Haar-like features [10], etc., and inputs the features into a classifier such as support vector machine (SVM) [11], AdaBoost [12], etc. [13–16]. However, these methods design features manually, the design process is subjective, there is a lack of theoretical guidance, and the generalization ability is poor [17,18]. With the development of deep learning, convolutional neural networks (CNNs) are widely used



because they can efficiently extract high-dimensional features of images and greatly improve detection accuracy [19]. In 2014, Girshick [20] proposed the region with a CNN (R-CNN) algorithm, using a selective search (SS) [21] strategy to determine candidate regions and applying CNNs to object detection. Compared with the best results of a traditional manual detection algorithm on the PASCAL Visual Object Classes Challenge (VOC) 2012 [22], the mean average precision (mAP) of the R-CNN algorithm was improved by more than 30%, reaching 53.3%. In 2015, Girshick [23] proposed the Fast R-CNN algorithm to improve detection speed by optimizing R-CNN. However, using the SS strategy to determine candidate regions takes a lot of time; thus, in 2016, Ren [24] proposed the Faster R-CNN algorithm, applying CNNs to the selection of candidate regions, and the region proposal network (RPN), which greatly improved the detection speed. Faster R-CNN has also become a mainstream method in the field of vehicle detection.

However, for the Faster R-CNN algorithm, host vehicle speed changes and preceding vehicle occlusions have a certain influence on preceding vehicle detection. In the process of generating candidate proposals, RPN adopts the anchor design, but Zhang [25] found that the mismatch of anchor size and small face reduced the detection accuracy of the small face, which is also suitable for small vehicles. Moreover, with changing host vehicle speed, the occurrence frequency of small, medium, and large vehicles may change, and the poor adaptation of anchor size to host vehicle speed also reduces the detection accuracy of small vehicles. Nonmaximum suppression (NMS) [26] is applied in the Faster R-CNN algorithm to perform redundant removal in vehicle detection. Although the traditional NMS selects the detection results with high scores and deletes adjacent results that exceed the threshold, the strict screening of NMS will lead to failure of occluded vehicle detection.

Optimizing anchors can improve the detection accuracy of the Faster R-CNN algorithm for the preceding vehicle. Wang et al. [27] matched the anchor size and receptive field to ensure that the network could obtain the appropriate number of vehicle features, improving the detection accuracy of Faster R-CNN on the KITTI dataset [28]. Ma et al. [29] chose anchor sizes that were object-adaptive and used self-adaptive anchors to enhance the structure of the Faster R-CNN algorithm, obtaining some success. Zhang et al. [30] improved the detection accuracy of small vehicles by adding a new anchor size of  $64 \times 64$  to the Faster R-CNN. Gao et al. [31] added two smaller scales ( $32 \times 32$ ,  $64 \times 64$ ) to the anchor box generation process to cover the high-frequency interval of the dataset between 30 and 60 pixels in width, and this method improved the Faster R-CNN performance on smaller vehicles. Generally, these methods are intended to make anchor size match vehicle size to achieve better detection of preceding vehicles, especially small vehicles, but they ignored a problem: With changing host vehicle speed, the occurrence of different sized vehicles will change, and anchors cannot adapt to this change.

Optimizing NMS can improve the performance of occluded vehicle detection. Bodla et al. [32] proposed a Soft-NMS algorithm with a penalty coefficient, which does not need to retrain the original model and can be easily integrated into any object detection algorithm using NMS, reducing the missed object detection rate. Zhao et al. [33] applied Soft-NMS to object detection, including vehicles, and proved that compared to NMS, the detection accuracy of the optimized Faster R-CNN could be improved by 1%–2% on the PASCAL VOC 2007. However, the above methods have a shortcoming: Soft-NMS is an optimized NMS algorithm by introducing the penalty coefficient of linear and Gaussian weighting, but it does not consider the impact of optimizing the penalty coefficients.

Thus, in this paper, we propose the speed classification random anchor (SCRA) and Q-square penalty coefficient (Q-SPC) methods to improve the detection accuracy of preceding vehicles when host vehicle speed changes and occlusion happens. The main contributions of this paper can be summarized as follows:

 The factor of speed is introduced: Through a real vehicle experiment, the vehicle speed is divided into three stages: 0–20, 20–60, and 60–120 km/h. The sizes of preceding vehicles at different speed stages are collected, and the k-means clustering algorithm [34] is used to analyze the rule of preceding vehicle sizes at different speed stages.

- 2. The relationship between the rule of preceding vehicle size and anchor size is established; at different speed stages, anchors are redesigned according to clustering results and random selection mode.
- 3. The detection results of preceding occluded vehicles are analyzed, the optimization requirements are put forward, and according to the requirements, the Q-SPC method is applied to optimize Soft-NMS.

This paper is organized as follows: First, the overall structure of the Faster R-CNN, RPN, NMS, and Soft-NMS algorithms are introduced. Second, according to the real vehicle experiment, the relationship between vehicle speed, preceding vehicle size, and anchor size is established; anchors are redesigned; and the SCRA method is put forward. Then, the Q-SPC method is proposed according to the detection results of occluded vehicles. Finally, in order to verify the effectiveness of the optimized algorithm, experimental verification is carried out.

# 2. Overall Structure of Faster R-CNN

#### 2.1. Structure of Faster R-CNN

The dominant paradigm in modern object detection is based on a two-stage approach: The first stage generates a sparse set of candidate proposals that should contain all objects while filtering out the majority of negative locations, and the second stage classifies the proposals into foreground/background classes and predicts proposal locations [35]. Faster R-CNN is a classical two-stage algorithm due to its efficient proposal extraction design. As shown in Figure 1, the overall structure of the Faster R-CNN algorithm is mainly divided into four parts: (1) Image feature extraction, corresponding to using VGG16 network model [36], a stack of basic convolutional (conv; 13 layers) + rectified linear unit (ReLU) activation function (13 layers) + pooling (4 layers) layers [37] is used to extract the input image features, and feature maps are used in the subsequent RPN and region of interest (RoI) pooling; (2) RPN: According to the feature maps input from the previous layer, RPN determines the object candidate proposals, which are the rough position of the object, including a rectangular box for proposals by the bounding box regression layer (reg layer), and the probability of object existence through the Soft-max function layer (cls layer) [38]; (3) RoI pooling: Proposal information and feature maps are collected, and proposals and feature maps are comprehensively extracted to get proposal features; (4) Classification and bounding box regression: According to the proposal features, the reg layer and cls layer are used to determine the object location and categories.



Figure 1. Schematic diagram of a faster region with a convolutional neural network (Faster R-CNN).

#### 2.2. Working Principle of RPN

RPN is a fully convolutional neural network. The RPN workflow in the test phase is shown in Figure 2. The previous layer of the input of this network is feature maps, and the output is rectangle proposals. First, a sliding window of  $n \times n$  (in this paper, n = 3) is performed on the feature map of the shared convolution network, new 512-dimensional feature maps are generated, and k regions are predicted on the original input image at the same time; these regions are anchors. Then the 512-dimensional feature maps are mapped to a low-latitude vector by  $1 \times 1$  convolution operation. This vector is used in two layers, the reg layer and the cls layer.

For the *k* anchors, the anchor point is located in the centroid of the sliding window, anchors are bounding boxes that have three different sizes ( $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ ) and three different aspect ratios (0.5, 1, and 2), thus *k* = 9, and the width and height of anchors are determined by Equation (1)

$$\begin{cases} W_{i,j} = 2^{s_j} \times round(\sqrt{(x \times y)/r_i}) \\ H_{i,j} = 2^{s_j} \times round(round(\sqrt{(x \times y)/r_i}) \times r_i) \end{cases}$$
(1)

where *W*, *H* represent the width and height of an anchor;  $r_i = (r_1, r_2, r_3) = (0.5, 1, 2)$ , and  $r_i$  indicates the aspect ratio;  $s_j = (s_1, s_2, s_3) = (3, 4, 5)$ , and  $2^{s_j}$  indicates the expansion factor of size; *round* stands for rounding; and *x*, *y* indicate the width and height of pixels of a feature point mapped back to the original input image, and x = y = 16.



Figure 2. Region proposal network (RPN) workflow in the test phase.

In the Faster R-CNN algorithm, any picture of  $P \times Q$  pixels will be adjusted to  $800 \times 600$  before being input to the CNN. Through the VGG16 model, the feature map of  $50 \times 38$  is obtained; one feature point corresponds to *k* anchors, and thus, there is a total of  $50 \times 38 \times k$  anchors on the input image. According to the cls layer and reg layer, each anchor obtains two scores on whether it contains an object (positive and negative) and four coordinates (horizontal and vertical coordinates of centroid, width, and height), and according to these parameters, all anchors are postprocessed, and finally, about 300 proposals are obtained. The postprocess steps are as follows:

1. According to coordinates, the position of each anchor is adjusted, the top 6000 position-corrected positive anchors are extracted based on confidence scores, the positive anchors whose range

exceeds the image are removed, and smaller anchors (width or height less than the threshold) are excluded.

2. Using NMS to handle the selected anchors, and based on the confidence scores, the top 300 anchors are selected as the proposals.

## 2.3. NMS and Soft-NMS

Faster R-CNN generates detection bounding boxes and scores for specific categories of objects. However, adjacent windows tend to have associated scores, which increases the false positives of the test results. To avoid such problems, the NMS algorithm is usually used to postprocess the detection bounding boxes.

The working process of the NMS is as follows: First, a series of boxes  $B_i$  and corresponding confidence scores  $S_i$  ( $i = 1, 2 \cdots j \cdots$ ) are generated. The highest confidence score  $S_j$  and corresponding box  $B_j$  are selected to determine the intersection over union (IoU) of the box  $B_i$  ( $i \neq j$ ) and  $B_j$ ; IoU can be obtained by Equation (2). Then, the confidence score of  $B_i$  ( $i \neq j$ ) is updated in accordance with Equation (3). If the confidence score of  $B_i$  is 0, remove the box, then select a box other than  $S_j$  and repeat the above operation until all the results are obtained:

$$IoU(B_i, B_j) = \frac{Area(B_i \cap B_j)}{Area(B_i \cup B_j)}$$
(2)

$$S_{i} = \begin{cases} S_{i} & \text{IoU}(B_{i}, B_{j}) < Threshold \\ 0 & \text{IoU}(B_{i}, B_{j}) \geq Threshold \end{cases}$$
(3)

Although NMS effectively reduces the false positives of the test results, this method is too crude; the detection bounding boxes adjacent to the highest confidence score box are forcibly cleared. If a real object appears in the overlapping area and has a large overlap, it will fail to detect and reduce the mAP of the algorithm. To solve this problem, Bodla [32] proposed the Soft-NMS algorithm, smoothed Equation (3), and proposed the penalty coefficients  $\lambda$  of linear and Gaussian weighting, which are shown in Equations (4) and (5), respectively; the confidence score formula is shown in Equation (6). By lowering the confidence score instead of directly deleting the box and removing the box whose confidence score is less than the set threshold, this method effectively improves the detection ability of Faster R-CNN for occluded objects:

$$\lambda = \begin{cases} 1 & \text{IoU}(B_i, B_j) < Threshold \\ 1 - \text{IoU}(B_i, B_j) & \text{IoU}(B_i, B_j) \ge Threshold \end{cases}$$
(4)

$$\lambda = e^{-\frac{IoU^2(B_i, B_j)}{\delta}} i \neq j$$
(5)

$$S_i = \lambda S_i \tag{6}$$

where *Threshold* = 0.3 and  $\delta$  = 0.3.

# 3. Problems in the Detection of Preceding Vehicles Using Faster R-CNN

#### 3.1. Problem of Preceding Vehicle Size and Host Vehicle Speed

In the course of vehicle running, there are generally multiple scales of preceding vehicles. To guarantee safe driving, the host vehicle should keep a safe distance from the front vehicle. Taking the widely used Mazda safe distance model [39] as an example, if the speed of the host vehicle increases, the safety distance should also increase. According to the working principle of the camera, vehicle size is negatively correlated with distance; vehicle size is small when distance is large [40]. This also means that with the increasing speed, the occurrence frequency of small vehicles also increases. To verify

this point, a real vehicle experiment was carried out. We divided vehicle speed into three stages and defined three vehicle sizes, then we collected the occurrence frequency of vehicles of different sizes. The details of real vehicle experiments are shown in the next section. In this section, we just show the frequency ratio chart to verify the above viewpoint.

As shown in Figure 3, at different speed stages, the occurrence frequency ratio of vehicles of different sizes is different; thus, the Faster R-CNN algorithm should take speed into account when detecting preceding vehicles.



**Figure 3.** Occurrence frequency ratio of vehicles of different sizes at different experimental vehicle speeds. With increasing speed, the occurrence frequency ratio of small vehicles increases significantly, while that of other types, such as large vehicles, decreases continually.

Moreover, Faster R-CNN has two stages: In the first stage, RPN generates anchors with three sizes ( $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ ) and three aspect ratios (1:1, 1:2, and 2:1), which creates a total of k = 9 anchors at each pixel of the feature map. The shapes of the anchors are invariant, and the anchor sizes do not match small vehicle sizes; they are too large to fit the small vehicles. Useful clues may be drowned in too many unnecessary clues; thus, the sizes of anchors should be adjusted to fit vehicle sizes.

The SCRA method is proposed to solve these problems.

# 3.2. Problem of Vehicle Occlusion

Although Soft-NMS improves the detection ability of the Faster R-CNN algorithm for occluded vehicles and has a good effect on vehicle detection, the optimization of Soft-NMS only introduces the penalty coefficient of linear and Gaussian weighting, without considering the impact of continuing to optimize the penalty coefficients. In this paper, the results of occluded vehicle detection were obtained, the method of Q-times multiplication of penalty coefficients was adopted to optimize the penalty coefficients, and the Q-SPC method is proposed to optimize the Soft-NMS.

# 4. Design of Optimized Faster R-CNN Algorithm

The working principles of the SCRA and Q-SPC methods are shown in Figure 4. The SCRA method was used to optimize anchors, and the Q-SPC method was used to optimize Soft-NMS.



**Figure 4.** Working principles of (**a**) the speed classification random anchor (SCRA) and (**b**) Q-square penalty coefficient (Q-SPC) methods.

As shown in Figure 4a, taking speed into account, we proposed the SCRA method to make anchor sizes fit vehicle sizes. The design steps of the SCRA method are as follows:

Step 1. The speed of the host vehicle is introduced, and according to the experiment, speed is divided into three stages: 0–20, 20–60, and 60–120 km/h. In order to make anchor sizes fit vehicle sizes, the rule of vehicle sizes should be analyzed at different speed stages. Thus, by collecting images and labels, the width (W) and height (H) of each vehicle are acquired.

Step 2. To analyze the rule of vehicle sizes at different speed stages, we employed the k-means clustering algorithm to deal with the width and height.

Step 3. Based on the results of clustering and postprocessing cluster centroids, maximum and minimum vehicle size, weighted mean value of cluster centroids, and extracted vehicle aspect ratios, anchors will be redesigned.

Step 4: Based on data such as cluster centroids and aspect ratios, too many anchors are generated; select anchors randomly to make the quantity reasonable.

Step 5: After finishing the above steps, the original anchors are updated with the redesigned anchors.

For our Q-SPC method, the design steps are as follows:

Step 1. The Faster R-CNN algorithm is used with NMS to deal with bounding boxes, because we want to optimize the Soft-NMS; thus, the NMS is replaced by Soft-NMS. Then the Faster R-CNN is used on the vehicle occlusion dataset, and the detection results are generated.

Step 2. Analyzing these results, some requirements are proposed. To satisfy requirements, the Q-times multiplication of penalty coefficients is adopted.

Step 3. Use optimized Soft-NMS to update Soft-NMS.

In the following sections, we introduce our SCRA and Q-SPC methods in detail.

4.1. Relationship between Anchors and Preceding Vehicle Size

4.1.1. The Rule of Preceding Vehicle Size at Different Speed Stages

In this section, in order to fit anchor sizes to vehicle sizes at different speed stages, we needed to divide the speed into stages and to explore the rule of vehicles size at different speed stages. Thus,

a real vehicle experiment was carried out to search reasonable speed stages and collect the images. The parameters of the actual vehicle experiment are shown in Table 1.

Experimental requirements	Parameters
Vehicle	Volkswagen
Transmission	Manual
Weather	Sunny
Road	Urban road/motorway
Camera	Sony XCL-C32C(CCD)
Acquisition time	10 hours
Maximum speed	120 km/h

Table 1. Parameters of real vehicle experiment.

Here, the frame rate of camera is 104 fps, the resolution is  $640 \times 480$ , and the camera is mounted behind the front windshield.

In our experiment, based on the status of the experiment vehicle and the speed limit conditions on urban roads and motorways, we chose four speed stages: 0–20, 20–60, 60–90, and 90–120 km/h (Table 2). When the vehicle exceeded 60 km/h, due to the high speed and large distance, the preceding vehicle size was small; moreover, when the vehicle exceeded 90 km/h, fewer vehicles were in front of the host vehicle, and the vehicle size was too small; in the process of labeling, vehicles had a similar size, and the width and height of each vehicle were almost the same (as shown in Figure 5). The rule of vehicle size may be unconvincing; thus, 90–120 km/h is not a suitable speed to analyze the rule of vehicle size. We grouped 60–90 and 90–120 km/h together, and divided the speed into 0–20, 20–60, and 60–120 km/h.

Table 2.	Status	of ex	perimental	and	preceding	vehicles a	at different	speed	stages
									()

Speed stage Experiment Vehicle Status	0–20 km/h	20–60 km/h	60–90 km/h	90–120 km/h
Transmission	First	Second to fourth	Fifth	Fifth
Phase	Starting/following Running		Running	Running
Number of preceding vehicles	More	More	Fewer	Much fewer
Environment	Urban road/traffic jam/intersection	General urban road	Good urban road/motorway	Motorway
Distance between preceding vehicles	Very small/small/large	Small/large	Large	Very large



**Figure 5.** Preceding vehicle sizes when experimental vehicle exceeded 90 km/h. Small vehicle sizes are similar, and the width (W) and height (H) of each vehicle are close.

Here, the definition of More, Fewer, and Much Fewer can be expressed by the mean of the number of preceding vehicles in per image; in this paper, we used 2.5, 1.5, and 0.5 to represent the approximate average values which correspond to More, Fewer, and Much Fewer, respectively.

In our experiment, we collected and screened images with a resolution of  $600 \times 480$  at different speed stages, then each image was labeled manually to obtain the width and height of preceding vehicles in the image; the types of selected vehicles are car and SUV. The parameters of the screened images are shown in Table 3.

Speed stage Parameters	0–20 km/h	20–60 km/h	60–120 km/h
Number	350	350	350
Number of preceding vehicles	764	744	596

Table 3. Parameters of screened images at different speed stages.

In this paper, we classify the preceding vehicle sizes as small, medium, large, and very large. Small, medium, and large are preceding vehicle sizes in the vehicle running experiment, and we added very large as a new size to fit some vehicles when the speed of the experimental vehicle is 0. As the k-means algorithm is a classical clustering algorithm which can set the number of clusters, and in order to corresponding to four vehicle sizes, k-means algorithm was adopted to deal with the width and height of preceding vehicles (K = 4). The clustering results at different speed stages are shown in Figure 6.



Figure 6. Clustering results of vehicle sizes: (a) at 0–20 km/h; (b) at 20–60 km/h; and (c) at 60–120 km/h.

However, it can be seen from the clustering results that there are too few vehicle sizes in cluster 4. Clusters 3 and 4 are recombined, and the centroids of clusters 3 and 4 are recombined by weighting. According to Equation (7), the centroid of new cluster A ( $x_{K=A}$ ,  $y_{K=A}$ ) is obtained. The cluster centroids and number of vehicle sizes per cluster at each speed stage are shown in Table 4.

$$\begin{cases} x_{K=A} = \frac{N_{K=3}}{N_{K=3} + N_{K=4}} x_{K=3} + \frac{N_{K=4}}{N_{K=3} + N_{K=4}} x_{K=4} \\ y_{K=A} = \frac{N_{K=3}}{N_{K=3} + N_{K=4}} y_{K=3} + \frac{N_{K=4}}{N_{K=3} + N_{K=4}} y_{K=4} \end{cases}$$
(7)

Among them, *x*, *y* represent the horizontal and vertical coordinates of cluster centroid, and *N* represents the number of vehicle sizes in each cluster.

Here, the horizontal and vertical coordinates of cluster centroids represent the width and height of preceding vehicles, respectively.

Speed stage Cluster (K)		0–20 km/h	20–60 km/h	60–120 km/h
1	Cluster centroid	(38,28)	(27,25)	(16,13)
	Number (N)	267	433	422
2	Cluster centroid	(75,50)	(60,45)	(40,29)
	Number (N)	260	217	111
3	Cluster centroid	(125,89)	(110,73)	(85,56)
	Number (N)	166	75	47
4	Cluster centroid	(210,130)	(205,138)	(160,98)
	Number (N)	77	19	16
А	Cluster centroid	(153,96)	(130,85)	(104,99)
	Number (N)	233	94	63

Table 4. Cluster centroids and number of vehicle sizes.

After recombination, the new cluster is obtained, and the vehicle sizes are redefined as small, medium, and large, corresponding to clusters 1, 2, and A, respectively. Each cluster is described by the mean of all values of this cluster (cluster centroid); thus, the cluster centroid is selected to describe the vehicle sizes that belong to this cluster. Fitting anchor sizes to vehicle sizes, as shown in Equation (1), vehicle size should include not only width and height, but also the aspect ratio (width/height); thus, the proportion of vehicle aspect ratios in clusters 1, 2, and A at different speed stages is collected to provide the basis for the design of anchors, and the results are shown in Figure 7.



**Figure 7.** Proportion of vehicle aspect ratios in clusters 1, 2, and A at different speed stages: (**a**) 0–20 km/h; (**b**) 20–60 km/h; and (**c**) 60–120 km/h.

## 4.1.2. Anchor Design Based on Vehicle Pixel Dimensions

In this section, without considering the aspect ratios, we defined the anchors as initial anchors, and after processing initial anchors by aspect ratio, we defined them as final anchors. At different speed stages, the occurrence frequency ratio of different vehicle sizes determines the number of initial anchors (*NIA*) belonging to the cluster, and the proportion of vehicle aspect ratios in each cluster determines the number of aspect ratios (*NAR*) belonging to the cluster. The number of final anchors corresponding to this cluster is *NIA* × *NAR*, and the number of final anchors corresponding to the stage is  $\sum_{K=1}^{A} (NIA_K \times NAR_K)$ ; K = 1, 2, A. At each speed stage, the design principles of *NIA* of each cluster are as follows: (1) The *NIA* should meet the occurrence frequency ratio of vehicle sizes shown in Table 5 as much as possible. (2) The *NIA* of each cluster should sum to 9 to meet the design of the original Faster R-CNN: *k* = 9. (3) The *NIA* should be larger than 1, so that the size of the anchor is variable to cover more vehicle sizes. The design principles of *NAR* of each cluster are as follows: (1) Compared with *NIA*, the ratio of *NIA* × *NAR* should be closer to the occurrence frequency ratios of vehicle sizes shown in Table 5. (2) Corresponding to the aspect ratios shown in Table 6.

Speed stage Vehicle Size	0–20 km/h	20–60 km/h	60–120 km/h
Small	0.35	0.58	0.7
Medium	0.34	0.29	0.19
Large	0.31	0.13	0.11

Table 5. Occurrence frequency ratios of vehicle sizes at different speed stages.

Here, ratios are obtained through the result of clustering shown in Table 4; at each speed stage, the ratio of each type (corresponding to cluster K) is  $N_K / \sum_{K=1}^A N_K, K = 1, 2, A$ .

Speed stage		0–20 km/h	20–60 km/h	60–120 km/h
Cluster (K)		•		
1	NIA	3	4	5
1	NAR	3	4	5
2	NIA	3	3	2
Ζ	NAR	3	3	3
<u>^</u>	NIA	3	2	2
A	NAR	3	3	3

Table 6. Values of number of initial anchors (NIA) and number of aspect ratios (NAR).

After obtaining the *NIA* and *NAR*, the vehicle sizes and aspect ratios should be selected. According to the acquired data at different speed stages, we established a method to select vehicle sizes. The method of selection and data chosen at different speed stages are shown in Figure 8. The selected vehicle sizes should satisfy the *NIA* condition and have a span to make anchor sizes more reasonable.



**Figure 8.** Method of selecting vehicle sizes. Weighted mean value of cluster centroids 1 and 2 is obtained by Equation (7).

• 0–20 km/h

When the vehicle speed is 0–20 km/h, the cluster centroid of clusters 1, 2, and A is (38, 28),(75, 50), and (153, 96), respectively, the minimum vehicle size of cluster 1 is (10, 9), the maximum vehicle size

of cluster A is (300, 200), corresponding to Figure 8, and the vehicle sizes corresponding to each cluster are selected:

$$\begin{pmatrix} [(w,h)_{K=1}] \\ [(w,h)_{K=2}] \\ [(w,h)_{K=A}] \end{pmatrix} = \begin{pmatrix} [(10,9) (38,28) (75,50)] \\ [(38,28) (75,50) (153,96)] \\ [(75,50) (153,96) (300,200)] \end{pmatrix}$$

The selected aspect ratios should satisfy the *NAR* condition and cover the aspect ratios shown in Figure 7 as much as possible. According to the proportion of different vehicle aspect ratios and *NAR* in each cluster, aspect ratios *R* corresponding to each cluster are selected:

$$\begin{pmatrix} [R_{K=1}] \\ [R_{K=2}] \\ [R_{K=A}] \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} 1 & 1.25 & 1.5 \\ 1 & 1.25 & 1.5 \\ \\ \begin{bmatrix} 1 & 1.5 & 2 \end{bmatrix} \end{bmatrix}$$

• 20-60 km/h

When vehicle speed is 20–60 km/h, the cluster centroid of clusters 1, 2, and A is (27, 25),(60, 45), and (130, 85), respectively, the weighted mean value of the centroid of clusters 1 and 2 is (38, 32), the minimum vehicle size of cluster 1 is (10, 7), the maximum vehicle size of cluster A is (300, 175), corresponding to Figure 8, and the vehicle sizes and aspect ratios *R* corresponding to each cluster are selected:

$$\begin{pmatrix} [(w,h)_{K=1}]\\ [(w,h)_{K=2}]\\ [(w,h)_{K=A} \end{bmatrix} \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} (10,7) & (27,25) & (38,32) & (60,45) \end{bmatrix} \\ \begin{bmatrix} (38,32) & (60,45) & (130,85) \end{bmatrix} \\ \begin{bmatrix} (60,45) & (300,175) \end{bmatrix} \end{pmatrix} \end{pmatrix} \begin{pmatrix} \begin{bmatrix} R_{K=1} \\ R_{K=2} \end{bmatrix} \\ \begin{bmatrix} R_{K=4} \end{bmatrix} \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} 1 & 1.2 & 1.4 & 1.6 \end{bmatrix} \\ \begin{bmatrix} 1 & 1.4 & 1.8 \end{bmatrix} \\ \begin{bmatrix} 1 & 1.5 & 2 \end{bmatrix} \end{pmatrix}$$

• 60–120 km/h

When the speed exceeds 60 km/h, the cluster centroid of clusters 1, 2, and A is (16, 13), (40, 29), and (104, 99), respectively, the minimum vehicle size of cluster 1 is (5, 6), the maximum vehicle size of cluster A is (202, 123), the weighted mean value of centroid of clusters 1 and 2 is (21, 16), the average of cluster centroid 1 and minimum vehicle size is (11, 10), corresponding to Figure 8, and the vehicle sizes and aspect ratios *R* corresponding to each cluster are selected:

$$\begin{pmatrix} [(w,h)_{K=1}]\\ [(w,h)_{K=2}]\\ [(w,h)_{K=A}] \end{pmatrix} = \begin{pmatrix} \begin{bmatrix} (5,6) & (11,10) & (16,13) & (21,16) & (40,29) \end{bmatrix} \\ & \begin{bmatrix} (21,16) & (104,99) \end{bmatrix} \\ & \begin{bmatrix} (21,26) & (122,123) \end{bmatrix} \\ & \begin{bmatrix} (21,26) & (122,126) \end{bmatrix} \\ & \begin{bmatrix} (21,26) & (122,126) \\ & \begin{bmatrix} ($$

Due to different clusters corresponding to different vehicle sizes, the anchor sizes are fit to vehicle sizes, so initial anchors correspond to different clusters as well. After selecting the width and height of vehicle sizes, the width (*IAw*) and height (*IAh*) of initial anchors are obtained by Equation (8):

$$(IAw, IAw)_K = (w, h)_K \tag{8}$$

After obtaining the width and height of initial anchors, aspect ratio R should be considered. The corresponding final anchor sizes are acquired; thus, Equation (1) is optimized as Equation (9), and the final anchor sizes at different speed stages are obtained by Equation (9):

$$\begin{cases} FAw_{S,K}^{i,j} = 1.25R_{S,K}^{j}round(\sqrt{(IAw_{S,K}^{i} \times IAh_{S,K}^{i})/R_{S,K}^{j}}) \\ FAh_{S,K}^{i,j} = 1.25round(\sqrt{(IAw_{S,K}^{i} \times IAh_{S,K}^{i})/R_{S,K}^{j}}) \end{cases}$$
(9)

where *S* represents the speed stage; *K* represents the *K*th cluster at *S*, and K = 1, 2, A; i, j represent the *i*th selected initial anchor size and the *j*th selected aspect ratio *R*, and  $i = 1_{S,K}, 2_{S,K}, \dots, NIA_{S,K}$ ,

 $j = 1_{S,K}, 2_{S,K}, \dots, NAR_{S,K}$ ; and FAw, FAh represent the width and height of the final anchor. In Faster R-CNN, the size of images is 800 × 600; in our experiment, the size of collected images is 640 × 480, so we multiplied by the expansion factor of 1.25.

## 4.2. Random Selection of Anchors

In order to adjust the number of anchors and improve the generalization ability of the training model, a random anchor structure is proposed in this paper. The purpose is as follows:

- 1. There are many repetitions in the final anchor sizes, so the random selection of anchor sizes is adopted to reduce repeatability.
- 2. Anchors are constantly changing to enhance the generalization ability of the training model.
- 3. Compared with k = 9, increasing the number of anchors will increase the proposals, cover more possible object areas, and improve the generalization ability of the training model.

Compared with the Faster R-CNN algorithm, we designed 27, 31, and 37 final anchors at the three speed stages, and the repeatability is strong. Therefore, in order to adjust the number of final anchor sizes and improve the generalization ability of the training model, a random selection model is proposed by Equation (10):

$$(FAw_{S,K}^{i}, FAh_{S,K}^{i}) = random^{j,m}(FAw_{S,K}^{i,j}, FAh_{S,K}^{i,j})$$

$$(10)$$

where *random*<sup>*j*,*m*</sup>() represents randomly extracted *m* data from *j* data. In this paper, m = 2, so at each speed stage, the number of final anchor sizes is 18, and the anchor sizes in the Faster R-CNN algorithm [41] are updated by final anchor sizes.

# 4.3. Q-square Penalty Coefficient

Compared with NMS, the Soft-NMS algorithm establishes a smoothing relationship between IoU and confidence score  $S_i$  and proposes the penalty coefficients  $\lambda$  of linear and Gaussian weighting; thus, the detection accuracy of occluded objects is improved by reducing the confidence score instead of strictly deleting the candidate boxes. The Soft-NMS algorithm was applied to the Faster R-CNN to replace NMS for detection of occluded preceding vehicles. From the detection results, there are several situations (as shown in Figure 9), and we optimized Soft-NMS from three aspects:

- 1. When vehicle occlusion is slight, occluded vehicles can be detected using the Soft-NMS algorithm; thus, we expect that the penalty intensity of our optimized Soft-NMS algorithm can be as consistent as possible with the original Soft-NMS algorithm, so the effect of detection remains as constant as possible.
- 2. When vehicle occlusion is serious, occluded vehicles cannot be detected using the Soft-NMS algorithm; thus, we expect that the penalty intensity of our optimized Soft-NMS algorithm can be stronger, the confidence score drops more sharply, and score ranking is lower, to reduce the influence of heavily occluded detection boxes.
- 3. When vehicle occlusion is between the above two conditions, some occluded vehicles can be detected using Soft-NMS, while some cannot be detected; thus, we explored the influence of adjusting penalty intensity on detection of occluded preceding vehicles.



Figure 9. Some situations of occluded vehicle detection. (a) When vehicle occlusion is slight, the occluded vehicle is detected; (b) when vehicle occlusion is serious, some heavily occluded vehicles cannot be detected; (c) when vehicle occlusion is moderate, the detection result is uncertain in the daytime; (d) when vehicle occlusion is moderate, the detection result is uncertain at night.

Therefore, to meet the above requirements, according to Equations (11) and (12), the Q-SPC method is proposed, and the  $\lambda^Q$  of linear and Gaussian weighting are obtained by Equations (11) and (12), respectively:

$$\lambda^{Q} = \begin{cases} 1 & \text{IoU}(B_{i}, B_{j}) < Threshold \\ (1 - \text{IoU}(B_{i}, B_{j}))^{Q} & \text{IoU}(B_{i}, B_{j}) \ge Threshold \end{cases}$$
(11)

$$\lambda^{Q} = \left(e^{-\frac{IoU^{2}(B_{i},B_{j})}{\delta}}\right)^{Q} i \neq j$$
(12)

where Q represents the times of multiplication; in this paper, Q = 1, 2, ..., 7, *Threshold* = 0.3, and  $\delta = 0.3$ .

Figure 10 shows the relationships between the penalty coefficient of linear weighting, Gaussian weighting, and the value of IoU, as well as the relationship between penalty coefficients and the power of Q. As shown in Figure 10a, when IoU < threshold,  $\lambda^Q = 1$  can make the penalty intensity stay in the same condition as Soft-NMS when occlusion is slight, while with increased IoU, the penalty intensity is stronger and the confidence score drops more sharply; and with increased Q, the intensity of punishment also changes. As shown in Figure 10b, when occlusion is slight, punishment is maintained as far as possible, but compared with linear weighting, the effect of preservation is not enough, and other effects are basically the same as linear weighting. Therefore, the method of multiplying the penalty coefficient by Q times can satisfy the requirements of optimized Soft-NMS.

0.8

Penalty coefficient 50 00 90

0.2

0 L 0

0.1 0.2 0.3 0.4 0.5 0.6



0

 $0.1 \ 0.2 \ 0.3 \ 0.4$ 

**Figure 10.** Relationships between penalty coefficient, intersection over union (IoU), and Q. (**a**) The relationship between penalty coefficient of linear weighting, IoU, and Q; (**b**) relationship between penalty coefficient of Gaussian weighting, IoU, and Q.

The optimized confidence score  $S_i$  is obtained by Equation (13):

(a)

0.7 0.8 0.9

$$S_i = S_i \times \lambda^Q \tag{13}$$

0.5 0.6 IoU

(b)

0.8

0.9

0.7

According to the confidence score, the code of Soft-NMS [42] is optimized, the possible object detection boxes are rescreened to improve the occluded vehicle detection accuracy of Faster R-CNN.

## 5. Experiments

In this section, we describe the experimental verification and comparison tests that were carried out to verify the effectiveness of the improved algorithm. Firstly, the evaluation indicators and training environment are introduced, then the datasets of the SCRA method at different speed stages and the Q-SPC are introduced. Finally, the different data training modes of the SCRA method are presented.

#### 5.1. Performance Evaluation Indicators

The most common model evaluation indicators (model performance) are mainly precision (P), recall (R), average precision (AP), and mean average precision (mAP). Precision represents the ability of the model to identify related objects; it is the percentage of correct predictions. Recall refers to the ability of the model to find all relevant objects; it is the percentage of true positives detected in all ground truths. For binary classification problems, AP measures the performance of the classifier, namely, the area of the P–R curve; AP is introduced to reflect the evaluation effect of the balance between precision and recall [43,44]. Precision and recall are determined by Equations (14) and (15), respectively:

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

where the meanings of *TP*, *FP*, and *FN* are shown in Table 7.

# Table 7. Interpretation of positive and negative examples.

Real result	Forecast result		
iteur result	Positive	Negative	
Positive	True positive (TP)	False negative (FN)	
Negative	False positive (FP)	True negative (TN)	

## 5.2. Training Environment and Datasets

The training parameters of our experiment are shown in Table 8.

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 (3 GB)
Training method	End-to-end
VGG model	VGG_CNN_M_1024
Learning rate	0.001
Number of iterations	70,000

Table 8. Training parameters.

In order to verify the effect of our design, training datasets were established by collecting images through a real vehicle experiment and labeling them manually. The parameters of our experimental datasets are shown in Tables 9 and 10. The datasets in Table 9 were used in the SCRA method; different datasets were collected at different speed stages. Training sets were used for training the algorithm, and test sets were used to test our training model. The proportion of training set and test set images is 70% and 30%, respectively. The datasets in Table 10 were used in the Q-SPC method, because vehicle occlusion usually occurs under low speed; thus, vehicle occlusion images were collected at 0–20 km/h in the real vehicle experiment.

#### Table 9. Datasets of the SCRA method.

	Speed (km/h)	Dataset	Training set	Test set
	0–20	1777	1244 (Data1)	533 (TestData1)
Number of	20-60	1326	928 (Data2)	398 (TestData2)
images	60-120	1371	960 (Data2)	411 (TestData3)
	Data integration (all)	4474	3132 (Data all)	1342 (TestData all)
Data training mode		Separate training	Ov	verall training

Table 10. Dat	aset of the	O-SPC	method.
---------------	-------------	-------	---------

	Speed (km/h)	Dataset	Training set	Test set
Number of images	0–20	1996	1397	599

# 5.3. Data Training Mode and Training

For the SCRA method, to verify the possible impact of the image data, as shown in Figure 11, data training of separate and overall training modes was adopted. For each training mode, redesigned anchors belonging to different speed stages were used to optimize Faster R-CNN; thus, there are three optimized Faster-RCNN algorithms corresponding to the three speed stages, Faster R-CNN with SCRA 1, Faster R-CNN with SCRA 2, and Faster R-CNN with SCRA 3, based on the data in Table 9. These algorithms were used for training; each optimized algorithm generated a detection model, Models 1, 2, and 3. Each model was used to test different test sets to verify the effect of this model.



(**a**) Separate training

(b) Overall training

**Figure 11.** Modes of data training: (a) Training set corresponding to different speed stages used for training the corresponding optimized algorithm; (b) training sets corresponding to different speed stages are assembled, and the assembled dataset is used for training.

For the Q-SPC method, because Soft-NMS optimization does not require retraining the model, based on the data in Table 10, the Faster R-CNN algorithm was used for training, and a detection model was generated. After modification, multiple tests can achieve verification of the Q-SPC method.

# 6. Results and Discussion

In Table 11, for the test sets belonging to different speed stages (TestData1, TestData2, and TestData3), the models with the highest AP values are Models 1, 2, and 3. With increased speed, the AP of both the optimized Faster R-CNN and Faster R-CNN algorithms declines, but compared to the Faster R-CNN algorithm, based on the SCRA method, the improvement in detection accuracy is 7.65%, 9.27%, and 15.14%, respectively.

Data training Mode	Detection Model _		AP of Test Sets (%)	)
		TestData1	TestData2	TestData3
Separate training	Model 0	78.98	71.07	61.24
	Model 1	86.63	61.42	52.36
	Model 2	43.95	80.34	55.85
	Model 3	37.97	69.61	76.38

Table 11. Test results of test sets under separate training modes.

Here, in order to make a comparison with our proposed algorithm, we selected the best detection result of nonoptimized Faster R-CNN for each test set, and we defined the acquired detection model as Model 0. Under separate training modes, we used Faster R-CNN to process the training and test sets belonging to same speed stage; under the overall training mode, Faster R-CNN was used to train all data, and the corresponding model 0 was used to test TestData all, TestData 1, TestData 2, and TestData 3.

The test results under overall training mode are shown in Table 12. For TestData1, TestData2, and TestData3, the models with the highest test results are Models 1, 2, and 3, and all AP values for overall training mode are higher than those for separate training modes. This means that the increased training set data samples have a positive impact on the detection accuracy, and compared to the Faster R-CNN algorithm, based on the SCRA method, the improvement in detection accuracy is 8.22%, 10.66%, and 16.11%, respectively.

Data Training Mode	Detection Model _	AP of Test Sets (%)				
		TestData1	TestData2	TestData3	TestData All	
Overall training	Model 0 Model 1 Model 2 Model 3	80.91 89.13 87.93 85.95	74.19 78.37 84.85 81.80	62.58 70.80 74.90 78.69	76.50 82.43 80.63 81.79	

Table 12. Test results of test sets under overall training mode.

The experimental results show the following:

- 1. For the test sets collected at different speed stages, no matter which data training mode we chose, the highest test results of TestData1, TestData2, and TestData3 are with Models 1, 2, and 3. That means that using the vehicle speed to adjust the anchor size adaptively can achieve the best detection effect.
- 2. By using the overall training mode, all the experimental results of Models 1, 2, and 3 were higher than with the Faster R-CNN algorithm. This proves that the rule of preceding vehicle size is effective and reasonable.
- 3. With increased vehicle speed, the occurrence frequency of small vehicles increased and the accuracy of the Faster R-CNN was gradually reduced, but the detection accuracy of the optimized algorithm was improved more obviously. This proves that the optimized anchors can match vehicle sizes, especially for small vehicles.
- 4. When preceding vehicle size was small and the color close to the background, sometimes there were some false positives of preceding vehicles detection.
- 5. Comparing separate and overall training, separate training modes had a poor detection effect on the test sets that do not belong to the corresponding speed stage, but under the overall training mode, the detection effect was better, and the generalization ability of overall training mode was better.

For occlusion of preceding vehicles, the Faster R-CNN algorithm was trained, and NMS, Soft-NMS based on linear weighting penalty coefficient and on Gaussian weighting penalty coefficient taking the influence of parameter  $\delta$  into account, and optimized Soft-NMS based on Q-SPC algorithm were applied to test separately, and the detection effect of occluded preceding vehicles is shown in Figure 12.



**Figure 12.** Detection effect of occluded preceding vehicles. Based on Equations (11)–(13), when Q = 1, Soft-NMS is nonoptimized, so when Q > 1, the result of Q = 1 is the comparison.

The following can be seen from the experimental results:

1. When Q = 1, compared with NMS, applying nonoptimized Soft-NMS (linear weighting, Gaussian weighting with  $\delta = 0.3$ , Gaussian weighting with  $\delta = 0.4$ ) for occluded preceding vehicles,

detection accuracy improved by nearly 2%, and the effect of Soft-NMS based on linear weighting penalty coefficient was slightly better than that of Gaussian weighting penalty coefficient.

- 2. With increased Q, detection accuracy increased continuously. For the optimized Soft-NMS with linear weighting, AP reached the maximum when Q = 4; for the optimized Soft-NMS with Gaussian weighting, AP reached the maximum when Q = 6, and  $\delta$  had little effect on the detection accuracy. On the whole, the effect of the linear weighting penalty coefficient was better than that of the Gaussian weighting penalty coefficient.
- 3. With the introduction of the Q-SPC method, for occluded vehicles, compared with the detection result when Q = 1, detection accuracy by Faster R-CNN improved 1%–2%, and the best Q values of linear and Gaussian weighting were 4 and 6, so this method has a certain effect.
- 4. According to our requirements, to optimize Soft-NMS, the ability of Gaussian weighting to maintain the penalty intensity was not enough, and this is one reason why the effect of Gaussian weighting was worse compared with linear weighting.

Figure 13 shows the test results based on the optimized Faster R-CNN algorithm using the SCRA method and the nonoptimized Faster R-CNN algorithm. Figure 13d shows the test results of occluded preceding vehicles based on the optimized Soft-NMS using the Q-SPC method and nonoptimized Soft-NMS.



**Figure 13.** Detection results of preceding vehicles by Faster R-CNN and our proposed optimized Faster R-CNN: (**a**) speed stage 0–20 km/h; (**b**) speed stage 20–60 km/h; (**c**) speed stage 60–120 km/h; (**d**)optimized Faster R-CNN; linear weighting was chosen and Q = 4.

# 7. Conclusions

In this paper, to improve the detection accuracy of preceding vehicles, an optimized Faster R-CNN algorithm based on SCRA and Q-SPC methods was proposed. Firstly, the reasons for degraded detection accuracy when the host vehicle speed increases were analyzed, and the factor of vehicle speed was introduced to redesign the anchors. Redesigned anchors can adapt to changes of preceding vehicle size when the host vehicle speed increases, and the SCRA method was proposed. Secondly, to achieve better performance on occluded vehicles, the Q-SPC method was proposed to optimize the Faster R-CNN algorithm. Finally, the experimental results showed that introducing the factor of host vehicle speed to make anchors adapt to vehicle size can bring 7%–17% accuracy improvement, and the method of Q-times multiplication of penalty coefficients can bring 1%–2% accuracy improvement for occluded vehicles. It was proved that the SCRA and Q-SPC methods have certain significance for improved accuracy of preceding vehicle detection.

In this paper, we improved the detection accuracy of preceding vehicles, and our method had no influence on detection speed, but Faster R-CNN is a two-stage algorithm, which causes this algorithm not to work in real time. In the next step, we will try to optimize the structure of Faster R-CNN to

improve the detection speed and try to extend our design to one-stage detection algorithms such as you look only once (YOLO) [45] or single shot multibox detector (SSD) [46]. Moreover, other types of vehicles such as buses and trucks and the influence of moving targets such as pedestrians and bikes were not considered. In the future, we will focus on diverse vehicle types and the impact of other road factors.

**Author Contributions:** Conceptualization, G.C. and S.W.; data curation, Z.L., Y.Y., and Q.W.; formal analysis, Y.W.; funding acquisition, S.W.; investigation, Z.L., Y.Y., and Q.W.; methodology, G.C.; project administration, S.W.; resources, G.C.; software, C.C.; supervision, S.W.; validation, S.W.; visualization, Y.W.; writing—original draft, G.C.; writing—review and editing, S.W.

**Funding:** This work was financially supported by the National Natural Science Foundation of China under Grant no. 71801144, and the Natural Science Foundation of Shandong Province under Grant no. ZR2019MF056.

Conflicts of Interest: The authors declare no conflict of interest.

# References

- Zhu, H.; Yuen, K.-V.; Mihaylova, L.; Leung, H. Overview of Environment Perception for Intelligent Vehicles. *IEEE Trans. Intell. Transp. Syst.* 2017, 18, 2584–2601. [CrossRef]
- 2. Kato, T.; Ninomiya, Y.; Masaki, I. Preceding vehicle recognition based on learning from sample images. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 252–260. [CrossRef]
- 3. Zhang, X.; Gao, H.; Guo, M.; Li, G.; Liu, Y.; Li, D. A study on key technologies of unmanned driving. *CAAI Trans. Intell. Technol.* **2016**, *1*, 4–13. [CrossRef]
- 4. Nguyen, V.D.; Nguyen, T.T.; Nguyen, D.D.; Lee, S.J.; Jeon, J.W. A Fast Evolutionary Algorithm for Real-Time Vehicle Detection. *IEEE Trans. Veh. Technol.* **2013**, *62*, 2453–2468. [CrossRef]
- Guo, J.; Wang, J.; Guo, X.; Yu, C.; Sun, X. Preceding Vehicle Detection and Tracking Adaptive to Illumination Variation in Night Traffic Scenes Based on Relevance Analysis. *Sensors* 2014, 14, 15325–15347. [CrossRef] [PubMed]
- 6. Yan, G.; Yu, M.; Yu, Y.; Fan, L. Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. *Optik Int. J. Light Electron. Opt.* **2016**, 127, 7941–7951. [CrossRef]
- 7. Han, J.; Liao, Y.; Zhang, J.; Wang, S.; Li, S. Target Fusion Detection of LiDAR and Camera Based on the Improved YOLO Algorithm. *Mathematics*. **2018**, *6*, 213. [CrossRef]
- 8. Han, G.; Su, J.; Zhang, C. A method based on Multi-Convolution layers Joint and Generative Adversarial Networks for Vehicle Detection. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 1795–1811.
- He, N.; Cao, J.; Song, L. Scale Space Histogram of Oriented Gradients for Human Detection. In Proceedings of the 2008 International Symposium on Information Science and Engieering, Washington, DC, USA, 20–22 December 2008; pp. 167–170.
- 10. Papageorgiou, C.P.; Oren, M.; Poggio, T. A general framework for object detection. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7–7 January 1998; pp. 555–562.
- 11. Burges, C.J. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* **1998**, 2, 121–167. [CrossRef]
- 12. Zhu, J.; Rosset, S.; Zou, H.; Hastie, T. Multi-class AdaBoost. Stat. Interface 2009, 2, 349-360.
- Teoh, S.S.; Bräunl, T. Symmetry-based monocular vehicle detection system. *Mach. Vision Appl.* 2012, 23, 831–842. [CrossRef]
- 14. Sivaraman, S.; Trivedi, M.M. Active learning for on-road vehicle detection: A comparative study. *Mach. Vision Appl.* **2014**, *25*, 599–611. [CrossRef]
- Kallenbach, I.; Schweiger, R.; Palm, G.; Lohlein, O. Multiclass object detection in vision systems using a hierarchy of cascaded classifiers. In Proceedings of the 2006 IEEE Intelligent Vehicles Symposium, Tokyo, Japan, 13–15 June 2006; pp. 383–387.
- 16. Son, T.T.; Mita, S. Car detection using multi-feature selection for varying poses. In Proceedings of the IEEE Conference on Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 507–512.
- 17. Liu, X.; Yang, T.; Li, J. Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network. *Electronics* **2018**, *7*, 78. [CrossRef]

- 18. Song, H.; Zhang, X.; Zheng, B.; Yan, T. Vehicle detection based on deep learning in complex scene. *Appl. Res. Comput.* **2018**, *35*, 1270–1273.
- 19. Meng, F.; Wang, X.; Shao, F.; Wang, D.; Hua, X. Energy-Efficient Gabor Kernels in Neural Networks with Genetic Algorithm Training Method. *Electronics* **2019**, *8*, 105. [CrossRef]
- Girshick, R.; Donahue, J.; Darrelland, T.; Malik, J. Rich feature hierarchies for object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
- 21. Uijlings, J.R.R.; Van De Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
- 22. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes(VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
- 23. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- 24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 91–99. [CrossRef] [PubMed]
- Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S. S<sup>3</sup>FD: Single Shot Scale-invariant Face Detector. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 192–201.
- 26. Neubeck, A.; Van Gool, L. Efficient Non-Maximum Suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR), Hong Kong, China, 20–24 August 2006; pp. 850–855.
- 27. Wang, Y.; Liu, Z.; Deng, W. Anchor Generation Optimization and Region of Interest Assignment for Vehicle Detection. *Sensors* **2019**, *19*, 1089. [CrossRef]
- 28. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* 2013, 32, 1231–1237. [CrossRef]
- 29. Ma, K.; Zhang, J.; Wang, F.; Tu, D.; Li, S.H. Fine-grained object detection based on self-adaptive anchors. In Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017; pp. 78–82.
- Zhang, Q.; Wan, C.; Bian, S. Research on Vehicle Object Detection Method Based on Convolutional Neural Network. In Proceedings of the 2018 International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018; pp. 271–274.
- Gao, Y.; Guo, S.; Huang, K.; Chen, J.; Gong, Q.; Zou, Y.; Bai, T.; Overett, G. Scale optimization for full-image-CNN vehicle detection. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 785–791.
- Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—improving object detection with one line of code. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5562–5570.
- 33. Zhao, W.; Yan, H.; Shao, X. Object detection based on improved non-maximum suppression algorithm. *J. Image Graph.* **2018**, *23*, 1676–1685.
- 34. Wong, M.A.; Hartigan, J.A. Algorithm AS 136: A K-Means Clustering Algorithm. J. R. Stat. Soc. Ser. C (Applied Stat.) 1979, 28, 100–108.
- Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
- Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations 2015 (ICLR2015), San Diego, CA, USA, 7–9 May 2015.
- 37. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. Nature 2015, 521, 436-444. [CrossRef] [PubMed]
- 38. Bishop, C.M. Pattern Recognition and Machine Learning; Springer: New York, NY, USA, 2006.
- 39. Seiler, P.; Song, B.; Hedrick, J.K. Development of a Collision Avoidance System. *SAE Tech. Pap. Ser.* **1998**. [CrossRef]
- 40. Wang, X.; Xu, L.; Sun, H.; Xin, J.; Zheng, N. On-Road Vehicle Detection and Tracking Using MMW Radar and Monovision Fusion. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2075–2084. [CrossRef]

- 41. Girshick, R. py-faster-rcnn. Available online: https://github.com/rbgirshick/py-faster-rcnn (accessed on 28 May 2019).
- 42. Bharat, S. soft-nms. Available online: https://github.com/bharatsingh430/soft-nms (accessed on 28 May 2019).
- 43. Huang, R.; Gu, J.; Sun, X.; Hou, Y.; Uddin, S. A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network. *Electronics* **2019**, *8*, 825. [CrossRef]
- 44. Xiang, X.; Lv, N.; Guo, X.; Wang, S.; El Saddik, A. Engineering Vehicles Detection Based on Modified Faster R-CNN for Power Grid Surveillance. *Sensors* **2018**, *18*, 2258. [CrossRef]
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- 46. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2016; pp. 21–37.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).