electronics

MDPI

Article

Part-Aware Refinement Network for Occlusion Vehicle Detection

Qifan Wang [1], Ning Xu [1], Baojin Huang [2,*] and Guangcheng Wang [2,*]

[1] School of Compute Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430072, China; wangqifan@whut.edu.cn (Q.W.); xuning@whut.edu.cn (N.X.)
[2] School of Compute Science, Wuhan University, Wuhan 430072, China
* Correspondence: huangbaojin@whu.edu.cn (B.H.); 2018102110030@whu.edu.cn (G.W.)

Abstract: Traditional machine learning approaches are susceptible to factors such as object scale, occlusion, leading to low detection efficiency and poor versatility in vehicle detection applications. To tackle this issue, we propose a part-aware refinement network, which combines multi-scale training and component confidence generation strategies in vehicle detection. Specifically, we divide the original single-valued prediction confidence and adopt the confidence of the visible part of the vehicle to correct the absolute detection confidence of the vehicle. That reduces the impact of occlusion on the detection effect. Simultaneously, we relabel the KITTI data, adding the detailed occlusion information of the vehicles. Then, the deep neural network model is trained and tested using the new images. Our proposed method can automatically extract the vehicle features and solve larger error problems when locating vehicles in traditional approaches. Extensive experimental results on KITTI datasets show that our method significantly outperforms the state-of-the-arts while maintaining the detection time.

Keywords: vehicle detection; partial confidence; occlusion

check for updates

Citation: Wang, Q.; Xu, N.; Huang, B.; Wang, G. Part-Aware Refinement Network for Occlusion Vehicle Detection. *Electronics* **2022**, *11*, 1375. https://doi.org/10.3390/electronics11091375

Academic Editor: Byung Cheol Song

Received: 21 March 2022
Accepted: 21 April 2022
Published: 25 April 2022

1. Introduction

Computer vision has been widely used in intelligent transportation fields [1–8] such as accurate identification of persons [9–11], reasonable allocation of traffic resources, and improvement of large-scale traffic linkage scheduling. Vehicle detection, as one of the important branches, provides basic support for the subsequent high-level tasks of vehicle decision planning and behavior control. Detecting vehicle type and location information from static images or dynamic videos is the main purpose of vehicle detection. Since 2005, vehicle detection in the traditional sense has been based on machine learning theory. First, vehicle features are extracted by Haar [12], HOG [13], SIFT [14], LBP [15] and other methods. Then, the extracted features are input to the support vector machine (SVM) [16], AdaBoost [17] and other classifiers for vehicle detection. Essentially, these methods rely on feature extraction models and are not effective for today's complex road conditions. Therefore, vehicle detection methods based on deep learning are produced in accordance with the needs of the times. Recent advances in vehicle detection are mainly attributed to the application of convolutional neural networks based on region selection and region classification. As a representative of the two-stage object detection method, the R-CNN family of algorithms [18–20] includes two stages of region selection and object confidence calculation, which greatly improves the efficiency of multi-object detection. Selective search algorithm [21] is usually used to extract candidate regions in region selection, but it cannot meet the efficiency requirements in real-time detection on mobile devices (CPU). The proposal of the YOLO family of algorithms [22–24] brings another breakthrough to object detection. It converts the two stages of region selection and classification into a single-stage regression problem, and realizes fast detection. On this basis, YOLO is applied to the detection of road vehicles to achieve real-time detection. To this end, Qiu et al. [25]

propose a vehicle recognition algorithm based on a convolutional neural network with fused edge features, which improves the recognition precision and the model's convergence speed through a simple and effective edge feature fusion method. Luo et al. [26] present a model based on Faster R–CNN with neural architecture search (NAS) optimization and feature enrichment to realize the effective detection of multi-scale vehicle targets in traffic scenes. NAS is a technology for automatically designing neural networks, which can automatically design high-performance network structures based on sample sets through algorithms. The method proposed by Haritha et al. deals with the detection of vehicles and is capable of handling distant small-scale vehicles without bothering the image scalability. Image scalability refers to whether it can be further analyzed (identified, etc.). However, these methods are only for small-scale vehicle detection and cannot be robust to occlusion scenes. Occlusion robustness refers to the model achieving good performance in occlusion scenarios. Therefore, it is urgent to propose an efficient and effective vehicle detection algorithm under occlusion conditions. Moreover, there exist lots of YOLO extensions purport to handle small objects and occlusion. Li et al. propose the addition of an attention mechanism, a CIoU (complete intersection over union) loss function, Soft-NMS (non-maximum suppression), and depthwise separable convolution to handle the occlusion. Du et al. present an improved YOLO model for infrared occlusion object detection under confusing background. Ryu et al. propose a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization. Most of these methods reduce the impact of occlusion on the model from the perspective of loss function and data augmentation, and our proposed method aims to achieve an occlusion-robust model from the network structure. Moreover, there exist many YOLO extensions that purport to handle small objects and occlusion. Li et al. [27] propose the addition of an attention mechanism, a CIoU (complete intersection over union) loss function, Soft-NMS (non-maximum suppression), and depthwise separable convolution to handle the occlusion. Du et al. [28] present an improved YOLO model for infrared occlusion object detection under confusing background. Ryu et al. [29] propose a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization. Most of these methods reduce the impact of occlusion on the model from the perspective of loss function and data augmentation, and our proposed method aims to achieve an occlusion-robust model from the network structure.

There are two main challenges for vehicle detection in actual scenarios: (1) The previous single confidence value based vehicle detection method uses the anchor box generation strategy to obtain the final detection box, which does not perform well for datasets with large object scale spans. (2) The output tensor adopts the predicted single value of the single-state network, which cannot describe the confidence score of the vehicle's components. For the occluded part, the normal single-state network [24] may assign low confidence to the occluded vehicle, resulting in missed detection and false detection. To cope with the above problems, we use the K-means clustering algorithm to generate a suitable anchor box size, adapt to the scale change of the dataset and improve the detection accuracy. Moreover, we optimize the model with the component confidence strategy, and use the confidence of the visible vehicle parts to correct the final detection confidence and improve the detection accuracy. In this paper, we choose the single-stage object detection method as the basic network structure. In order to eliminate the influence of scale and occlusion factors on vehicle detection, the KITTI dataset with our relabeling is used to train the deep neural network model. Combined with multi-scale training and component confidence generation strategy, the parameters are adjusted reasonably to transform the vehicle detection problem. For the multi-classification problem of vehicles, its efficiency and accuracy are further improved.

In brief, the major contributions of this paper are highlighted as follows.

- We propose a part-aware refinement network for occlusion vehicle detection, which optimizes the model with the component confidence strategy, and uses the confidence of the visible vehicle parts to correct the final detection confidence. Our proposed method can automatically extract the vehicle features, it has solved the problems of larger error when locating vehicles in traditional approaches, and improved the vehicle detection recall.

- we adopt the K-means clustering algorithm to generate a suitable anchor box size, adapt to the scale change of the dataset and improve the detection accuracy.
- we relabel the KITTI dataset, adding the detailed occlusion information of the vehicles. The new dataset can provide better supervision for occluded vehicle detection.

2. Proposed Method

2.1. Preliminary Knowledge on Single-Stage Object Detection

The algorithm proposed in this paper is based on a single-stage object detection framework to adapt to real-time vehicle detection in real-world scenarios. Thus this subsection gives preliminary knowledge on single-stage object detection. The single-stage object detection algorithm omits the selection steps of candidate regions, simplifies the network structure, and dramatically improves the detection speed of the algorithm. However, it meanwhile leads to insufficient feature extraction, sacrificing a certain degree of accuracy, and the appearance of detection objects.

In 2016, Redmond et al. proposed the YOLOv1 [22] for the first time, which inputs the raw image into the neural network to directly obtain the bounding box and classification results. Specifically, YOLO first divides the image into S × S squares. If the center point of an object falls in a square, N bounding boxes are predicted around the square, and one calculates the confidence of the object surrounded by each bounding box. Finally, the non-maximum suppression algorithm (NMS) [30] is used to remove the redundant bounding box of a single object, and the detection result is obtained. Compared with other algorithms, the biggest advantage of this algorithm is that the detection speed is fast. YOLOv1 can detect 45 frames per second while distinguishing the background and the object to be measured well. However, it loses a certain accuracy and the detection performance of small objects.

In order to solve the problem of low detection accuracy of the YOLOv1, YOLOv2 [23] adds BN (batch normalization) after each convolutional layer, applies multi-anchor training methods to increase the number of detectable objects. Moreover, YOLOv2 is fine-tuned on a pre-trained CNN using a high-resolution dataset. YOLOv3 [24] introduces a residual module and nine anchor boxes for detection, which improves the detection accuracy of small targets while ensuring the detection speed. Specifically, YOLOv3 first transforms the input image to a size of N × N (N is 416 by default), and uses Darknet-53 to extract image features. Then, the transformed image is divided into S × S equal-sized grids (S is 13, 26, and 52, respectively, representing three scales), and the network randomly selects a new image size every 10 batches. For each grid, three anchor boxes generate three bounding boxes, respectively. as shown in Figure 1, for each grid, the convolutional neural network predicts 4 values to represent each bounding box, denoted as t_x, t_y, t_w, t_h, ie the coordinates (x, y) and the width w and height h of the object. If the object center is offset from the upper left corner of the image in the grid, the offset is assumed to be (c_x, c_y), the height and width of the anchor box are recorded as P_w and P_h. The bounding box is corrected as:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h}. \end{aligned} \tag{1}$$

During training, the sum of squared errors is used to calculate the sum of the losses, including the localization loss and the confidence loss. Assuming that the real position coordinate is $\hat{t}^*$, the loss function is minimized by the method of gradient descent, where the gradient is the difference between the real value and the predicted value of the coordinate $\hat{t}^* - t^*$. For the confidence calculation of the predicted object, YOLOv3 converts it into a regression problem. First, the overlap rate between the anchor box and the real object bounding box is calculated [31], denoted as A_{iou}, and A_{iou}^{max} is assumed to represent all anchor boxes and the real object bounding box. The maximum value of the overlap ratio, and then the probability P_{object} that anchor box contains the object is calculated as follows:

$$P_{\text{object}} = \begin{cases} 1 & A_{iou} \geq A_{iou}^{\max} \\ 0 & A_{iou} > 0.5 \quad \text{and} \quad A_{iou} < A_{iou}^{\max} \end{cases} \tag{2}$$

YOLOv3 assigns only one anchor box to each object. In particular, those anchor boxes that are considered to contain no objects are considered to have a loss value of 0. In order to enable YOLOv3 also to perform multi-label classification of an object, YOLOv3 simultaneously uses the binary cross-entropy loss [32] method for class prediction.

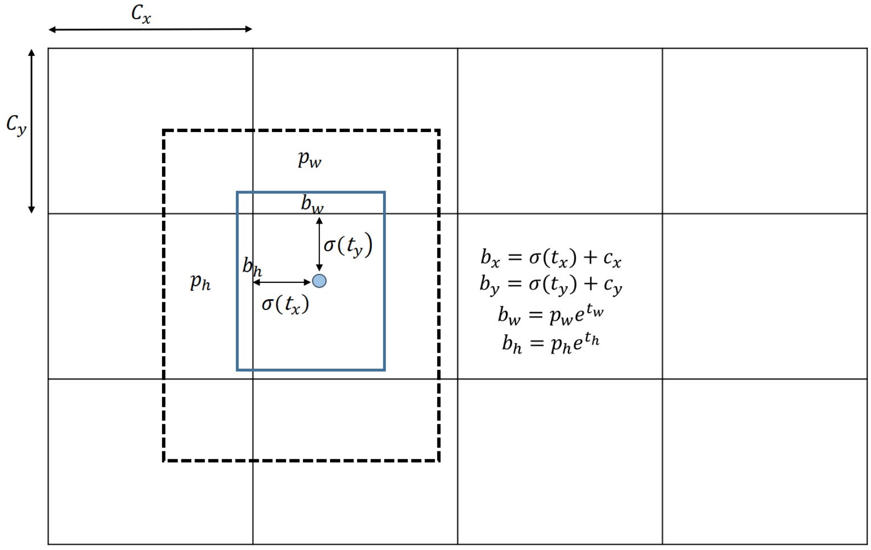


Figure 1. Bounding boxes with dimension priors and location prediction.

YOLOv4 adopts a better Mish activation function and introduces the SPP module to further improve the detection effect. YOLOv5 was proposed almost at the same time as the YOLOv4, which is slightly weaker than YOLOv4 in performance, but much stronger than the latter in flexibility and speed.

2.2. *Part-Aware Refinement Network*

2.2.1. Output Tensor Overview

One-stage vehicle detection can be viewed as a regression problem and generates tensors to predict the output. As shown in Figure 2, the width (W) and height (H) of the output tensor depend on the number of grids of the input image, and the depth (K) depends on the number of anchor boxes per grid. The prediction output definition of each anchor box is shown in Figure 3. The box offset of the model is defined by the position and scale between the bounding box $x_{gt}, y_{gt}, w_{gt}, h_{gt}$ and the matching anchor box x_i, y_i, w_k, h_k, where $i \in [1, W], \quad j \in [1, H], \quad k \in [1, K]$. The scale parameter (δ_w, δ_h) describes the scale difference between the prediction box and the anchor box:

$$\delta_{w,(ijk)} = \log\left(\frac{w_{gt}}{w_k}\right), \quad \delta_{h_i(ijk)} = \log\left(\frac{h_{gt}}{h_k}\right). \tag{3}$$

For the position parameter (δ_x, δ_y), it corresponds to the relative position of the upper left corner in the prediction grid, and its boundary is [0, 1):

$$\delta_{x,(ijk)} = \sigma\left(\frac{x_{gt} - x_i}{w_{grid}}\right), \delta_{y,(jjk)} = \sigma\left(\frac{y_{gt} - y_i}{h_{grid}}\right). \tag{4}$$

where σ is the sigmoid function.

The single-stage vehicle detection method defines the confidence level of object existence in Equation (5). Then, the conditional probability of the class C object is defined in Equation (6). Finally, the final object probability is obtained by multiplying the object conditional probability by the object confidence.

$$c_{(ijk)} = P_{(ijk)}(\text{Object}) \times \text{IOU}^{gt}_{(ijk)} \tag{5}$$

$$p_{m,(ijk)} = P_{(ijk)}(\text{Class} = m \mid \text{Object}), m \in [1, C] \tag{6}$$

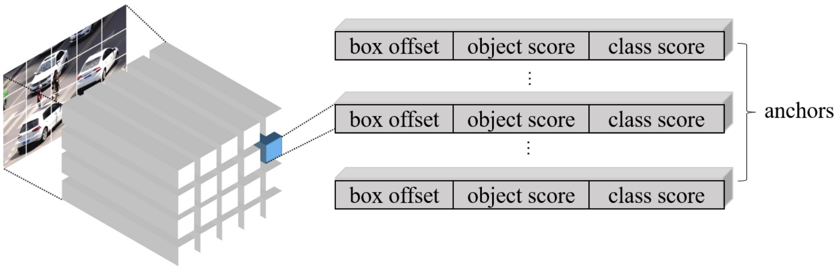
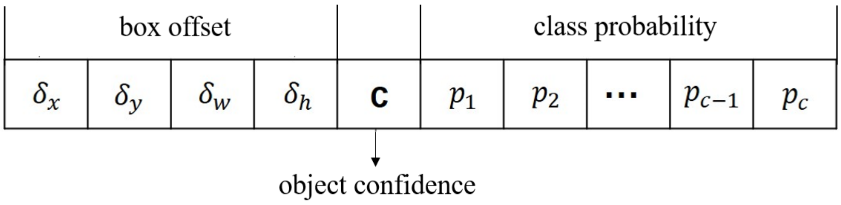


Figure 2. Structure of the single-stage object detection output tensor.



box offset					class probability				
δ_x	δ_y	δ_w	δ_h	**C**	p_1	p_2	···	p_{c-1}	p_c

object confidence

Figure 3. Output format per anchor.

2.2.2. Part-Aware Refinement Network

Our key idea for occlusion handling is to block the prediction confidence rather than represent it as a single value for an existing single-state network. A normal single-state network might assign low confidence to an occluded vehicle due to the occluded part, but our model can use the confidence of the visible parts of the vehicle to correct the final detection confidence of the vehicle.

We first introduce the concept of a part confidence map [33,34] represented by V, which is an $M \times N$ grid (generated by the sigmoid function) in the range [0, 1], as shown in Figure 4. The ground truth of the part confidence map is generated as follows. We identify a bounding box for the overall vehicle and divide it into an $M \times N$ grid. The YOLO algorithm determines whether it is an object box by the confidence of several prediction boxes. Our method aims to convert the confidence of each prediction box from a single value to a confidence map, on this basis, the confidence of the occluded part can be weakened to improve the vehicle prediction box. For each cell (m, n), $m \in [1, M]$, $n \in [1, N]$, set $V_{gt}(m, n) = 1$ if the cell area occupied by the vehicle exceeds τ_v times. In our experiments, we set $M = 3$, $N = 6$ and $\tau_v = 0.5$.

For occlusion processing, we expand the output tensor to include predictions of the part confidence map $\hat{V}$ (see Figure 4). That is, the network predicts the $\hat{V}$ detection output, from which we compute the final occlusion-aware detection score for each anchor box. We adopt an end-to-end learnable scoring method [35], as shown in Figure 5. We define P blocks $W_p \in R^{M \times N}$, $p \in [1, P]$. We compute the intermediate part score by element-wise dot product prediction of the part confidence map $\hat{V}$:

$$S_p = \sum_{m=1}^{M} \sum_{n=1}^{N} \left(\hat{V}(m, n) \cdot W_p(m, n) \right). \tag{7}$$

With $S = [s_1, s_2, \ldots, s_p]$, we compute the final vehicle prediction confidence through a multilayer perceptron (single hidden layer and ReLU layer):

$$S_{car} = \sigma\left(w_2^T \max\left(0, w_1^T s\right)\right). \tag{8}$$

$\{W\}_{p=1}^{P}$ and $w_{1,2}$ are parameters that need to be learned, where P represents the number of modes that occlude vehicles. We choose $P = 40$ through experiments. The mode indicates the occlusion situation, such as occlusion above the vehicle, occlusion below, etc. Setting 40 modes means that there are 40 occlusion situations. At the same time, we tried different multi-layer perceptron structures and found that the effect in Equation (8) is the best.

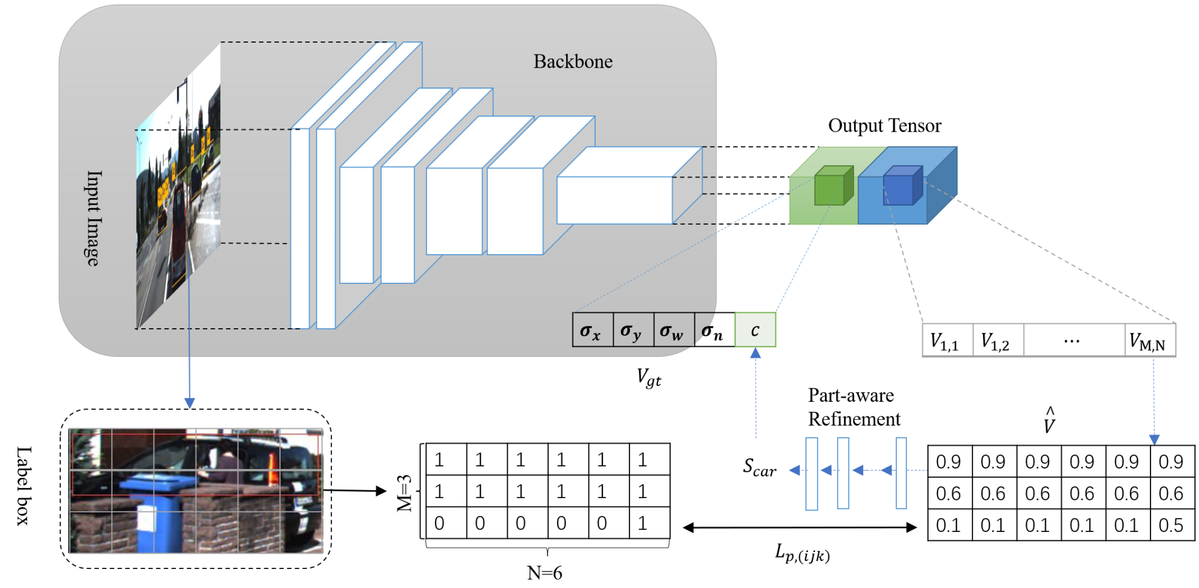


Figure 4. The overview of our occlusion handling method.

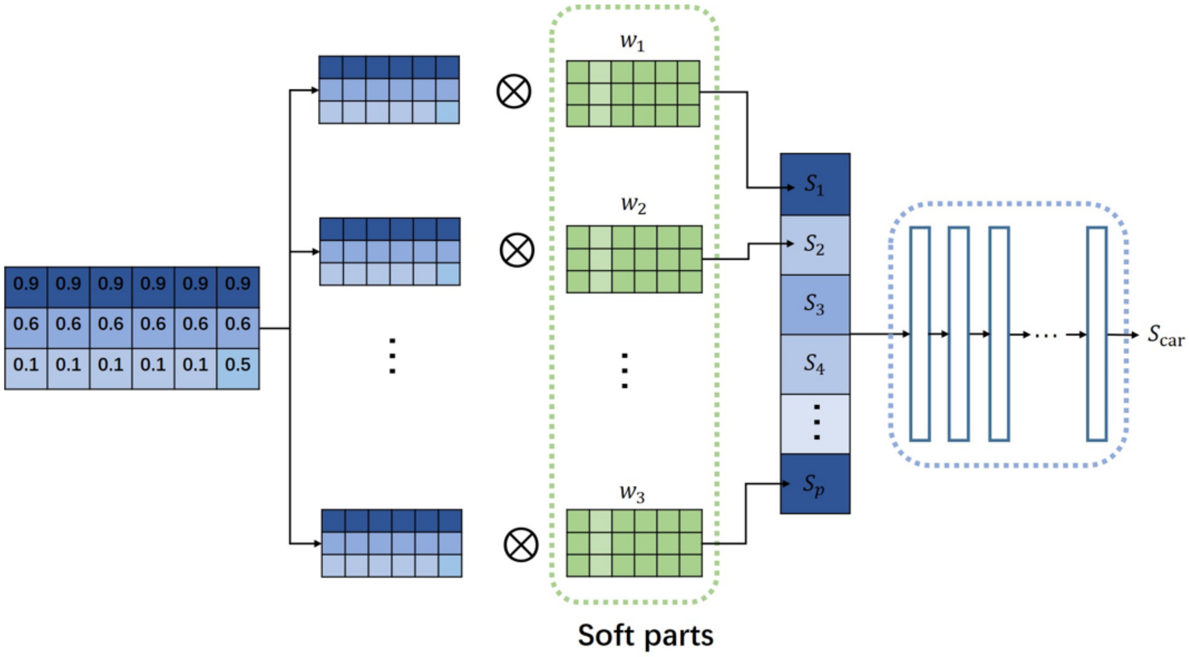


Figure 5. Structure of part-aware refinement network.

2.2.3. Loss Function

The loss of the one-stage model consists of two parts: the localization loss L_l and the confidence loss L_c. After introducing the part confidence map, we define the part confidence loss L_p as follows:

$$L_p = \left(\lambda_p^+ H^+_{(ijk)} + \lambda_p^- H^-_{(ijk)}\right) \times \sum_{m=1}^{M}\sum_{n=1}^{N}\left(V_{(ijk)}(m,n) - \hat{V}_{(ijk)}(m,n)\right)^2 \tag{9}$$

where $H^+_{(ijk)} = 1$ indicates that the (ijk) anchor box is a positive example but is occluded, and $H^-_{(ijk)} = 1$ is a negative example, thus the final loss value is the weighted sum of 3 loss values:

$$L = \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\left(\lambda_l L_{l,(ijk)} + \lambda_c L_{c,(ijk)} + \lambda_p L_{p,(ijk)}\right), \tag{10}$$

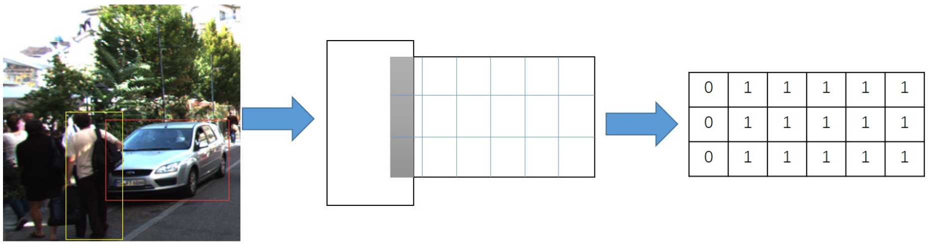
where λ_l, λ_c and λ_p are hyperparameters to trade-off the loss.

3. Experimental Results

3.1. KITTI Dataset

The KITTI dataset [36] is the original dataset for training in this paper, which contains real image data collected under a variety of complex scenes (highways, urban streets, rural streets, etc.). Each of these images includes up to 15 vehicles with different motion states (stationary or moving) and different degrees (truncated or occluded). In the experiment, 5000 images with different characteristics were randomly selected and converted into COCO dataset format for training the model. At the same time, 2500 images are selected to test the trained model to verify the effectiveness of the algorithm.

The definition of occlusion in the KITTI dataset is 0 (unoccluded), 1 (partial occlusion), 2 (large area occlusion), and 3 (unknown). In order to get the true value of the part confidence in Equation (9), we preprocess the KITTI dataset here, as shown in Figure 6. For an image in the KITTI test set, we find the overlapping part of its annotation boxes, and determine which annotation box is occluded by the occlusion degree of each annotation box. Then, for the occluded bounding box, set $V_{gt}(m,n) = 0$ in the overlapping part, and set $V_{gt}(m,n) = 1$ in the non-overlapping part. This generates part confidence for each bounding box.



0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

Figure 6. KITTI dataset preprocessing.

3.2. Implementation Details

The evaluation criteria of the validity test in the vehicle detection problem include recall and precision. The recall and precision used in this paper are in the range of [0, 1]. The precision is for the prediction result, it indicates how many of the predicted positive samples are true positive samples. Then, there are two possibilities for the prediction to be positive, one is to predict the positive class as the positive class (TP), and the other is to predict the negative class as the positive class (FP). That is

$$\text{precision} = \frac{TP}{TP + FP}. \tag{11}$$

The recall is for the original sample, it indicates how many positive examples in the sample are predicted correctly. There are also two possibilities, one is to predict the original positive class as a positive class (TP), and the other is to predict the original positive class as a negative class (FN). The calculation method is as follows:

$$\text{recall} = \frac{TP}{TP + FN}. \tag{12}$$

In this paper, all experiments are conducted on Intel(R) Core(TM) i5-6500 CPU@3.20 GHz, 64 G RAM, TITAN X GPU. Every network is trained for 100 epochs, and the batch size is adjusted to 16 according to the GPU and network parameters. The gradually decreasing learning rate during the training process will be more accurate than the fixed learning rate. Therefore, this paper adopts the strategy of adjusting the learning rate at equal intervals. After each epoch, the learning rate is adjusted by 0.90 times, and the initial learning rate is set to 0.001. We choose YOLOv3 as our baseline model.

3.3. Multi-Scale Anchor Box Design

We use the K-means clustering algorithm to get the anchor box size, where the similarity is defined as the overlap ratio of the rectangular boxes (represented by R_{IOU}), and annotate the road vehicle target training set. The distance function of K-means clustering is

$$d(S,T) = 1 - R_{IOU}(S,T). \tag{13}$$

Among them, S and T represent the size and center of the rectangular frame of the object, respectively, and $R_{IOU}(S,T)$ represents the overlap ratio of the two rectangular frames. After weighing the average R_{IOU} and the number of anchor boxes, as shown in Figure 7, take 15 anchor boxes, which are (10, 95), (13, 36), (19, 156), (22, 51), (29, 80), (34, 255), (39, 56), (44, 112), (60, 82), (64, 298), (74, 136), (108, 177), (119, 279), (175, 293), (175, 521). Each grid at each scale predicts five bounding boxes with five anchor boxes.

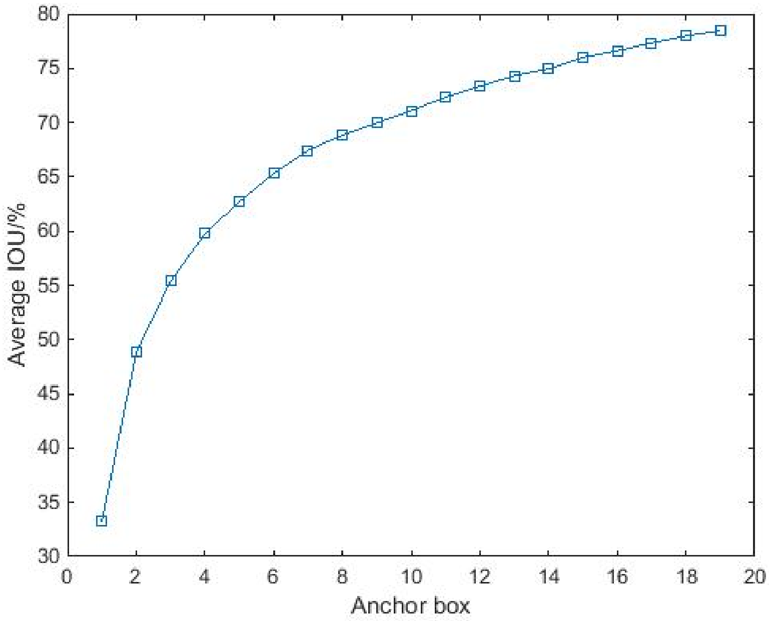


Figure 7. The relationship between average IOU and the number of anchor boxes.

As shown in Figure 7, ideally, the larger the number of anchor boxes, the better. However, the number of anchor boxes affects the execution efficiency of the algorithm. We observed that when the number of anchor boxes reaches 15, we will continue to increase the number of anchor boxes. The improvement of the average intersection ratio is not obvious, so the average intersection ratio and the number of anchor boxes are weighed, and 15 anchor boxes are selected.

3.4. Ablation Studies

We adopt different strategies to train and test the neural network respectively, which further verifies the effectiveness of different strategies proposed in this paper. As shown in Table 1. It can be found that the detection precision is improved by 0.78% with the change of the number of anchor boxes. The reason is that there are nine anchor boxes in the original network, and vehicles with smaller sizes are not easy to be detected. Comparing strategies A and C in Table 1, it can be seen that the strategy of component confidence generation has obvious advantages. It solves the problem of vehicles being occluded in the road and improves the detection precision by 1.62 percentage points. This experiment fully proves the effectiveness of using multi-scale training and part confidence generation strategy to train the model, and improves the detection precision and recall of the model.

Table 1. Detection results before and after algorithm training strategy improvement.

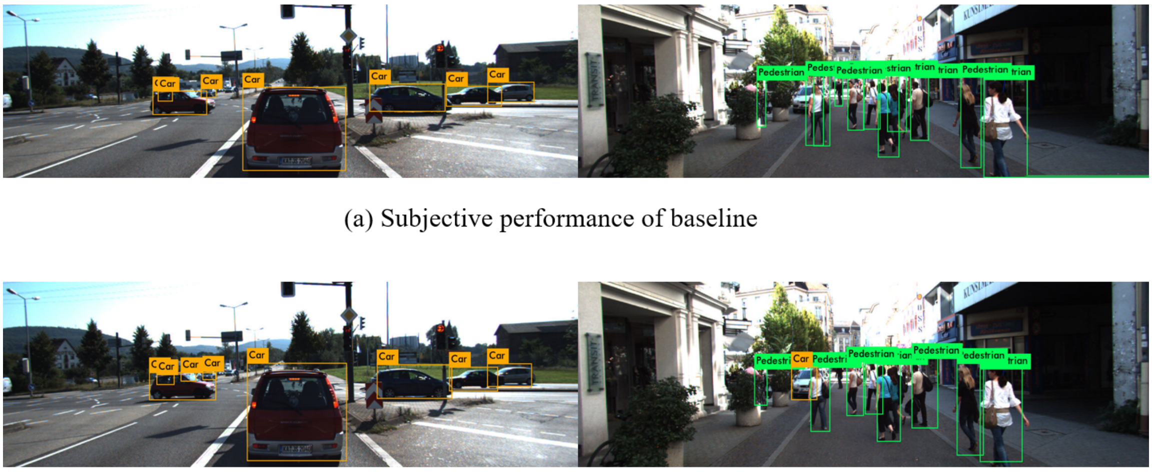
Model	Number of Anchors	Part-Aware Refinement	Recall (%)	Precision (%)
A	9	✗	94.24	84.24
B	16	✗	94.56	85.42
C	9	✓	96.12	86.26
D	16	✓	97.09	87.52

3.5. Compare the Detection Performance of the Two Algorithms in the Same Scene

The baseline model and our proposed method were trained using the KITTI dataset, and tested through the test set (only vehicle indicators were extracted). The precision and recall of the two algorithms are shown in Table 2, and the detection effect is shown in Figure 8.

Table 2. Detection effect of our method and baseline model in the same scene.

Method	Recall (%)	Precision (%)	Average IOU
Baseline	94.24	84.64	72.38
Ours	**97.09**	**87.52**	**78.03**



(a) Subjective performance of baseline

(b) Subjective performance of our method

Figure 8. Comparison of detection effects between baseline and our method.

From the data comparison in Table 2, it can be seen that the recall rate and precision of our model through multi-scale training and part confidence generation strategy are better than the original baseline model. Compared with the baseline model, the time performance is still good, the recall is increased by 2.85%, the precision is increased by 2.88%, and the average IOU is increased by 5.65%. As can be seen from Figure 8, our model can detect smaller-scale vehicles and partially occluded vehicles with good results. For sparse vehicle and dynamic pedestrian objects, the difference between our method and the two algorithms of the baseline model is small. For the case where some vehicles and pedestrian objects overlap, the original baseline model has missed detection, and our method can detect overlapping objects.

3.6. Comparison with SOTA Methods

We use the KITTI dataset to train SSD [37], RFCN [38], Faster R-CNN [20], YOLOv3 [24], FE-CNN [25], RIAC [26], MVD [39] and our method respectively. The accuracy and detection time are then evaluated on the testing set, as shown in Table 3. SSD combines the methods of Faster R-CNN and YOLOv1 to improve the detection accuracy. RFCN adopts the method of Faster R-CNN. Compared with Faster R-CNN, it improves the detection speed and reduces the amount of calculation. YOLOv3 converts the detection into a regression problem, the network structure and detection speed are improved, but the detection accuracy is relatively low. FE-CNN is based on a convolutional neural network with fused edge features, which improves the recognition precision and the model's convergence speed through a simple and effective edge feature fusion method. RIAC is based on Faster R–CNN with NAS optimization and feature enrichment to realize the effective detection of multi-scale vehicle objects in traffic scenes. MVD deals with the detection of vehicles and is capable of handling distant small-scale vehicles without bothering the image scalability.

The advantage of this paper is that it combines the part confidence generation strategy and multi-scale training method to improve the accuracy based on YOLOv3. It can be seen from Table 3 that the method in this paper improves the accuracy while ensuring the faster detection rate of the YOLO series algorithm itself. It is more suitable for real-time detection systems. In summary, the method in this paper is an effective vehicle detection method.

Table 3. Precision and time comparison between our method and existing algorithms for detecting single vehicle images.

Method	**SSD [37]**	**RFCN [38]**	**Faster R-CNN [20]**	**YOLOv3 [24]**	**FE-CNN [25]**	**RIAC [26]**	**MVD [39]**	**Ours**
Precision (%)	83.54	88.39	86.30	84.64	85.25	86.19	85.92	87.52
Recall (%)	94.93	97.62	96.34	94.24	94.87	96.55	96.14	97.09
IOU	74.39	76.11	75.99	74.26	72.38	76.11	75.96	78.03
Time (s)	0.177	0.281	0.293	0.082	0.083	0.151	0.097	0.091

4. Conclusions

This paper proposes a part-aware refinement network for occlusion vehicle detection, which combines multi-scale training and part confidence generation strategies to transform the vehicle detection problem in complex environments into an easy-to-implement regression problem. The model is trained on the KITTI dataset, and the parameters are adjusted reasonably. Experiments verify that for detection scenes of different scales and different degrees of occlusion, the trained models can extract vehicle features well and have good robustness.

Author Contributions: Conceptualization, Q.W.; methodology, Q.W. and N.X.; software, B.H. and G.W.; validation, N.X., B.H. and G.W.; formal analysis, N.X., B.H. and G.W.; investigation, Q.W.; resources, Q.W., N.X., B.H. and G.W.; data curation, Q.W. and N.X.; writing—original draft preparation, Q.W.; writing—review and editing, B.H. and G.W.; visualization, Q.W. and N.X.; supervision, N.X.; project administration, N.X.; funding acquisition, N.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The experiment uses a public dataset.

Acknowledgments: The authors would like to thank the reviewers and the editor for their careful reviews and constructive suggestions to help us improve the quality of this paper.

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

References

1. Chang, L.; Chen, Y.T.; Wang, J.H.; Chang, Y.L. Modified YOLOv3 for Ship Detection with Visible and Infrared Images. *Electronics* **2022**, *11*, 739. [CrossRef]
2. Jiang, X.; Gao, T.; Zhu, Z.; Zhao, Y. Real-time face mask detection method based on YOLOv3. *Electronics* **2021**, *10*, 837. [CrossRef]
3. Liu, C.; Wu, Y.; Liu, J.; Sun, Z. Improved YOLOV3 network for insulator detection in aerial images with diverse background interference. *Electronics* **2021**, *10*, 771. [CrossRef]
4. Wang, G.; Wang, Z.; Gu, K.; Jiang, K.; He, Z. Reference-free dibr-synthesized video quality metric in spatial and temporal domains. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 1119–1132. [CrossRef]
5. Jiang, K.; Wang, Z.; Yi, P.; Chen, C.; Han, Z.; Lu, T.; Huang, B.; Jiang, J. Decomposition makes better rain removal: An improved attention-guided deraining network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3981–3995. [CrossRef]
6. Jiang, K.; Wang, Z.; Yi, P.; Chen, C.; Wang, Z.; Wang, X.; Jiang, J.; Lin, C.W. Rain-free and residue hand-in-hand: A progressive coupled network for real-time image deraining. *IEEE Trans. Image Process.* **2021**, *30*, 7404–7418. [CrossRef]
7. Jiang, K.; Wang, Z.; Yi, P.; Lu, T.; Jiang, J.; Xiong, Z. Dual-path deep fusion network for face image hallucination. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 378–391. [CrossRef]
8. Jiang, K.; Wang, Z.; Yi, P.; Wang, G.; Gu, K.; Jiang, J. ATMFN: Adaptive-threshold-based multi-model fusion network for compressed face hallucination. *IEEE Trans. Multimed.* **2019**, *22*, 2734–2747. [CrossRef]
9. Wang, Z.; Jiang, J.; Yu, Y.; Satoh, S. Incremental re-identification by cross-direction and cross-ranking adaption. *IEEE Trans. Multimed.* **2019**, *21*, 2376–2386. [CrossRef]
10. Wang, Z.; Jiang, J.; Wu, Y.; Ye, M.; Bai, X.; Satoh, S. Learning sparse and identity-preserved hidden attributes for person re-identification. *IEEE Trans. Image Process.* **2019**, *29*, 2013–2025. [CrossRef]
11. Wang, Z.; Hu, R.; Chen, C.; Yu, Y.; Jiang, J.; Liang, C.; Satoh, S. Person reidentification via discrepancy matrix and matrix metric. *IEEE Trans. Cybern.* **2017**, *48*, 3006–3020. [CrossRef] [PubMed]
12. Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 1, pp. 900–903.
13. Negri, P.; Clady, X.; Hanif, S.M.; Prevost, L. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP J. Adv. Signal Process.* **2008**, *2008*, 782432. [CrossRef]
14. Ma, X.; Grimson, W.E.L. Edge-based rich representation for vehicle classification. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, pp. 1185–1192.
15. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face recognition with local binary patterns. In Proceedings of the European Conference on Computer Vision, Prague, Czech Republic, 11–14 May 2004; pp. 469–481.
16. Teoh, S.S.; Bräunl, T. Symmetry-based monocular vehicle detection system. *Mach. Vis. Appl.* **2012**, *23*, 831–842. [CrossRef]
17. Cao, X.; Wu, C.; Yan, P.; Li, X. Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2421–2424.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, USA, 7–12 December 2015; Volume 28.
21. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
22. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
23. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
24. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
25. Qiu, L.; Zhang, D.; Tian, Y.; Al-Nabhan, N. Deep learning-based algorithm for vehicle detection in intelligent transportation systems. *J. Supercomput.* **2021**, *77*, 11083–11098. [CrossRef]

26. Luo, J.Q.; Fang, H.S.; Shao, F.M.; Zhong, Y.; Hua, X. Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment. *Def. Technol.* **2021**, *17*, 1542–1554. [CrossRef]
27. Li, Y.; Li, S.; Du, H.; Chen, L.; Zhang, D.; Li, Y. YOLO-ACN: Focusing on small target and occluded object detection. *IEEE Access* **2020**, *8*, 227288–227303. [CrossRef]
28. Du, S.; Zhang, B.; Zhang, P.; Xiang, P.; Xue, H. FA-YOLO: An Improved YOLO Model for Infrared Occlusion Object Detection under Confusing Background. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1896029. [CrossRef]
29. Ryu, S.E.; Chung, K.Y. Detection Model of Occluded Object Based on YOLO Using Hard-Example Mining and Augmentation Policy Optimization. *Appl. Sci.* **2021**, *11*, 7093. [CrossRef]
30. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 850–855.
31. Rosebrock, A. Intersection over Union (IoU) for Object Detection. Diambil Kembali Dari PYImageSearch. 2016. Available online: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection (accessed on 1 July 2021).
32. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
33. Mathias, M.; Benenson, R.; Timofte, R.; Van Gool, L. Handling occlusions with franken-classifiers. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1505–1512.
34. Tian, Y.; Luo, P.; Wang, X.; Tang, X. Deep learning strong parts for pedestrian detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1904–1912.
35. Noh, J.; Lee, S.; Kim, B.; Kim, G. Improving occlusion and hard negative handling for single-stage pedestrian detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 966–974.
36. Fan, Q.; Brown, L.; Smith, J. A closer look at Faster R-CNN for vehicle detection. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 124–129.
37. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–14 September 2016; pp. 21–37.
38. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016; Volume 29.
39. Haritha, H.; Thangavel, S.K. A modified deep learning architecture for vehicle detection in traffic monitoring system. *Int. J. Comput. Appl.* **2021**, *43*, 968–977. [CrossRef]