Pattern Analysis and Applications (2021) 24:1635–1647
https://doi.org/10.1007/s10044-021-01009-4

THEORETICAL ADVANCES

Vehicle object detection method based on candidate region aggregation

Luyang Zhang[1] · Haitao Wang[1] · Xinyao Wang[1] · Qiang Liu[1] · Huaibin Wang[1] · Hailong Wang[2]

Received: 27 September 2020 / Accepted: 7 July 2021 / Published online: 24 July 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
Multi-scale vehicle detection is an important application in the field of object detection, and Feature Pyramid Network (FPN) is an important means to deal with multi-scale object detection tasks. However, baseline method is the common method used in most of the existing network structure, which represents the input image information by selecting one from the output layer of FPN, and discard other layers. This not only limits the performance of the network structure, but also performs poorly when dealing with the problem of excessive scale differences. To solve this problem, a novelty candidate region aggregation network (CRAN) is proposed in this paper. The candidate regions of different feature layers are effectively aggregated to improve the network generalization performance. Specifically, calculate the similarity between different feature layers through a feature quality score module, and use this as a quantity factor to determine the number of candidate regions reserved for the corresponding feature layer. Finally, they are aggregated into a more comprehensive candidate region group. Further, in order to improve the detection efficiency of small objects, an area cross entropy loss function is proposed. It makes the model pay more attention to small targets by adding a monotonic decrease based on the area. Finally, the proposed CRAN and the area cross entropy loss function are applied to the advanced detectors. The experimental results in the KITTI and UA-DETRAC datasets show that this method has good performance on vehicle objects in different scenarios, and can meet the requirements of practical application.

Keywords FPN · CRAN · Area cross entropy · Quality score module · Vehicle detection

1 Introduction

Vehicle object detection plays an important role in intelligent transportation system (ITS). It is the prerequisite and foundation for follow-up research work such as vehicle recognition, vehicle tracking, and traffic statistics [1]. With the rapid development of deep neural networks, object detection has become a major research hotspot in computer vision tasks. General object detection has achieved great success, driven by the deep convolutional neural network (DCNN). Object detection refers to the combination of object segmentation and recognition, which can not only detect the position of the object in the picture or video, but also recognize the category of the object. It is widely used in intelligent transportation systems, intelligent monitoring systems, military target detection, and medical imaging field. However, vehicle detection still faces many challenges in complex traffic scenes, such as various lighting conditions, occlusion, and low-resolution [2].

Nowadays, many scholars at home and abroad are committed to the research of object detection and have obtained good results. The proposed architecture can be divided into two categories: two-stage detectors [3–7], and one-stage detectors [8–10]. The two-stage detector achieves better detection accuracy, but sacrifice speed and consume resources. The one-stage detector has poor detection accuracy but is more efficient in the training and inference process, and it is more suitable for real-time detection in real scenes.

In order to detect objects of different scales, CNN-based target detection algorithms adopt multi-scale outputs [11–13]. Among, YOLO v3 and Mask R-CNN use the Feature Pyramid Network (FPN) [14] idea to fuse feature

✉ Haitao Wang
875727548@qq.com

[1] College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China

[2] Jiangsu Urcat Wall Computer System Co., Ltd., Nantong 226000, Jiangsu, China

maps of adjacent scales through the concat method. FPN uses a top-down, side-to-side connection method to fuse the features of two adjacent scales. The high-resolution feature map contains more fine-grained features of the object, and the low-resolution features contain more contour information. Effective feature aggregation can improve network performance.

With the introduction of FPN, choosing a suitable FPN output layer has become a problem that must be solved. The traditional method is based on the Region of Interest (RoI) obtained by the RPN to select, which is based on the width w and height h of the RoI, using the formula (1) proposed by [14] to find the best k layer as a sample.

$$k = \left\lfloor k_0 + \log_2 \left(\sqrt{wh/m} \right) \right\rfloor \tag{1}$$

where m represents the size of the pre-trained ImageNet input picture, and k_0 represents the level corresponding to the RoI with an area of $w \times h$. However, we think that the choice of a single-layer FPN may limit the ability of network description. [15] proved our idea, which achieved better detection accuracy than the baseline method by summing the candidate regions generated by all the feature layers. However, this summation method inevitably increases the complexity of the network, and the training process requires more resources. This is understandable, because the summation method increases the number of candidate regions by 5–6 times or even more, and requires a lot of computing resources.

Inspired by [15], a candidate area aggregation network (CRAN) is proposed in this paper. First, re-extract features of FPN output features through a convolutional layer; then, a quality score module is constructed to calculate the similarity between different feature layers. The similarity result is used as a quantitative factor to determine the number of candidate regions in the corresponding feature layer. Finally, a more comprehensive set of candidate regions is generated. Since our quantity factor is derived from the output feature map of FPN, and each group of feature maps is derived from the same input image. Therefore, the proposed quality score module can be applied to any input picture. In addition, in order to solve the problem of difficult detection of small targets in the process of vehicle detection, an area cross entropy loss function is proposed. This paper designs a monotonically decreasing function based on the area of the candidate region to add weight to the cross entropy loss function. Our intuition is that small goals should be assigned more weight, while big goals require less weight. The introduction of area cross entropy loss is beneficial to the detection of small targets and the improvement of the performance of the model.

The main contributions of this paper are as follows:

1) A novelty candidate region aggregation network (CRAN) is proposed to effectively aggregate candidate regions of feature layers of different scales. Improve the performance of the network structure to handle multi-scale problems.
2) An area cross entropy loss function is proposed to improve the detection performance of the model for small targets. In this paper, each candidate region is assigned a different weight during the classification process, and the weight depends on the area of each generated candidate region.
3) The proposed CRAN and area cross entropy loss are introduced into the current advanced detectors and tested on challenging datasets.

The flow of the remaining paper is as follows. Object detection architecture and feature fusion method are elaborated in Sect. 2. Section 3 describes the proposed approach. In Sect. 4, the experimental setup, benchmark datasets and experimental results are presented. The conclusions are placed in Sect. 5.

2 Related work

2.1 Object detection

With the increasing popularity of intelligent transportation systems, many experts have begun to study vehicle object detection [16]. There have been many outstanding studies in the early days, such as Harr [17], SIFT [18], HOG [19], DPM [20, 21]. However, traditional detection algorithms require manual acquisition of relevant target feature information, which results in high complexity and a large amount of redundancy. Severely affects the running speed and is difficult to realize engineering in real scenarios. With the development of deep learning, especially the proposal of deep learning algorithms based on convolutional neural networks, object detection has entered an intelligent development stage. Through parameter sharing and sparse connection, the object detection algorithm can avoid the complicated process of manually extracting features. It effectively solves the problems of poor portability and missing features of traditional models [22]. In addition, with the rapid development of GPU technology, the computing speed of deep learning has also shown an exponential increase.

Recently, CNN-based two-stage and one-stage detectors are continuously updating object detection performance in several benchmark datasets. The first is a two-stage architecture based on R-CNN [3, 23, 24]. In order to improve the training efficiency of the network, in 2015, *HE* et al. [5] proposed Faster R-CNN, which designed an RPN network to generate proposals under a unified framework (Fig. 1).

Then, a series of excellent two-stage detectors appeared, and trying to optimize the network architecture [25–27], training strategy [28, 29], adding auxiliary modules [30–32] to improve the network.

In 2016, the yolo method was proposed by Redmon et al. [8]. The candidate bounding box regression, and classification are directly integrated into the same convolution network, and obtained extremely fast detection speed. However, due to the rough network design, it is far from reaching the accuracy requirements of real-time target detection, and there are problems such as inaccurate target positioning, poor detection of small objects and multiple objects. Subsequently, Redmon and others continued to improve the YOLO algorithm and proposed YOLO v2 [33] and YOLO v3 [11], respectively. At the same time, Liu et al. [9] proposed asingle-shot detector (SSD), which combines the regression idea of the YOLO model and the anchor mechanism of faster R-CNN. SSD surpasses Faster RCNN in detection speed and accuracy, but it does not consider the correlation between different layers and different scale targets, resulting in poor detection of small objects. Then, RSSD and DSSD were proposed, and the performance was greatly improved.

2.2 Multi-scale features

As the limitation of single feature representation becomes more and more prominent, people began to study multi-feature fusion technology in order to find a better feature representation. The existing feature fusion methods can be divided into two types: direct addition to average and weighted sum. In fact, the former is a special form of the latter. Many scholars have tried this research [16, 34–36] and achieved good performance. However, in the field of object detection, we often need to deal with targets of different scales, so multi-scale issues must be considered. [14] proposed a feature pyramid network (FPN) to perform feature representation from different levels, and has been proven to be effective for general object detection. However, the selection of the FPN output feature layer is based on heuristic, which limits its performance to a certain extent. On this basis, a novel candidate region aggregation network is designed to effectively utilize all the output layer information of FPN and improve the performance of the network structure.

2.3 Classification loss function

In terms of object classification, the cross entropy loss function adjusts network parameters by describing the distance between vectors. It has always had a good performance and is used in many advanced algorithms [3, 4, 6, 7]. However, we can see from the expression of the cross entropy loss function that its weight parameter for all input samples is 1, which makes it perform poorly in dealing with complex problems, such as a serious imbalance in the number of samples (1: 100), the object size gap is too large, etc. Based on this problem, [8] proposed Focal loss for the first time. Focal loss effectively solves the problem of imbalance in the sample category ratio by adding a balance coefficient to the cross-entropy. In the process of vehicle detection, the detection of small targets and low-resolution targets has always been a challenging problem. Therefore, based on the idea of area weight loss in [37], we propose a cross entropy loss function based on the area factor. By assigning more weights to small targets and a small amount of weights to large targets, the problem of excessive object size gaps in the vehicle detection process is effectively solved.

3 Methodology

In this section, the candidate region aggregation network (CRAN) and area cross entropy loss function are described in detail.

3.1 Candidate region aggregation network (CRAN)

The proposal of FPN effectively solves the problem of multi-scale feature selection, and is an architecture that can select appropriate features according to the size of the image. FPN effectively solves the problem of multi-scale feature selection, and can select the appropriate feature architecture according to the size of the image. Many literatures have proved that FPN has the ability to maintain effective spatial information, and avoids the complicated calculation problems caused by the refinement of features at each scale. In the network architecture, the selection of feature maps is generally based on the baseline method to select one of them as the input of the RoI layer. Although the baseline method is a more general method, but the [15] proves that the baseline method is similar to the random selection method, and has proved this idea through experiments. The experiment selected some samples from the COCO data set, and the baseline method, random method and direct sum method were selected for comparison experiments. Figure 2 shows the progress of the training process. It can be seen that the progress of the random method and the baseline method are relatively similar, and the average accuracy gap is small. It shows that each output feature map of FPN contains valid information, and it is not comprehensive to use any single feature map to represent the input image. In addition, the experiment also directly sums the output feature maps of FPN. The results show that the training progress of the summation method is basically consistent with the baseline method, and after the 9th epoch, the test accuracy exceeds the result of the baseline method. The above experimental

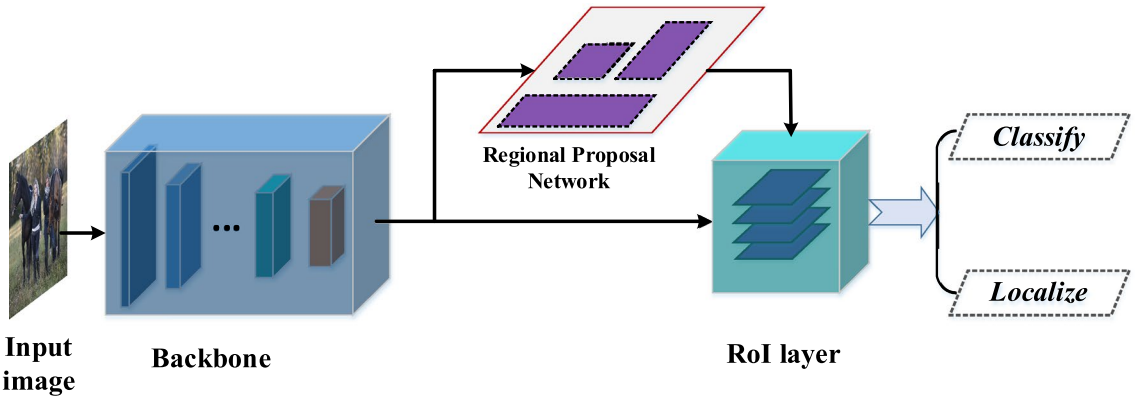

Fig. 1 Two-stage detector architecture

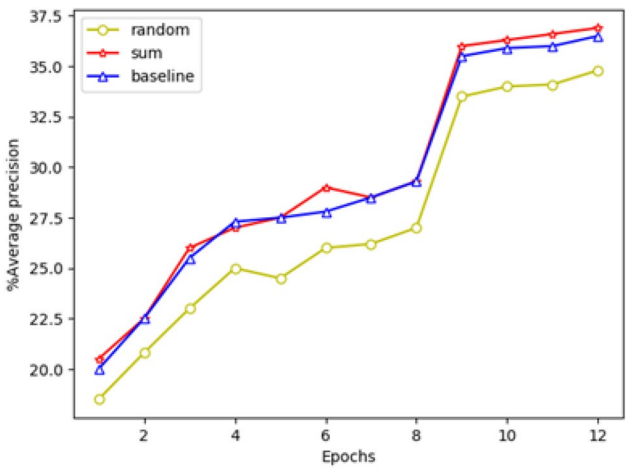

Fig. 2 The average prediction accuracy of different FPN layersselected under the COCO data set

results show that every output feature map of FPN cannot be ignored. Therefore, effectively aggregating the output feature maps of FPN is of great help in improving the performance of the network model.

Based on the above problems, this paper proposes a candidate region aggregation network (CRAN). Our inspiration is that although the method of summation can increase the richness of candidate regions, it greatly increases the consumption of computing resources, and a large number of candidate regions are easy to cause interference between classes. Therefore, this article tries to process the generated candidate regions to minimize the number while ensuring the richness. CRAN mainly consists of three modules: feature re-extraction module, quality score module and aggregation module. The network structure is shown in Fig. 3.

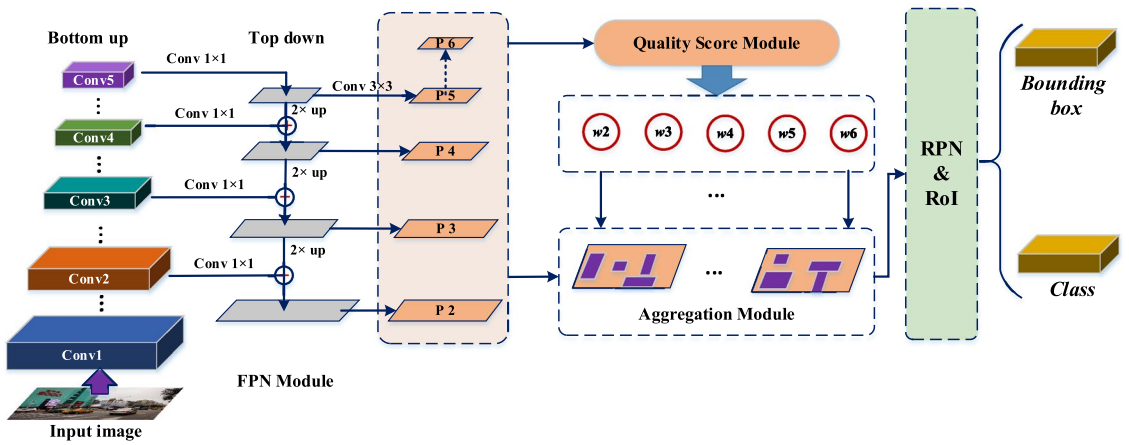

Fig. 3 Candidate region aggregation network

3.1.1 Feature re-extraction module

Through the ResNet50 backbone network and FPN, feature maps P2–P6 of different scales can be extracted from the input image, where P6 is obtained by up-sampling from P5. Since the above features are obtained by superimposing the up-sampling features and the basic features C2–C5, in order to better fuse these two features, this paper re-extracts the merged features P2–P6. The output features of FPN are re-extracted using a convolution kernel with a size of 3×3, and the [38] proves that this method can effectively improve the quality of features.

3.1.2 Quality score module

The quality score module is mainly to learn the quantity factor of FPN output feature maps by introducing an attention mechanism. We tried two ways to learn quantity factor to explore them: one is based on Feedforward Neural Network (FNN), the other is based on Convolutional Neural Network (CNN) method.

The based on FNN method first converts feature maps of different scales into the same size according to the principle of forward propagation. Then calculates the degree of similarity between the corresponding feature maps of the baseline method and other feature maps. Finally, the feature quality score is determined through the normalization operation. The basic structure is shown in Fig. 3, P_k is the feature map of the K_{th} layer selected by the baseline method, P_i is the FPN output feature map, and P_i^* is the result of P_i tiled expansion. It is worth noting that the vector sizes corresponding to different object sizes are different. In order to ensure that the vector similarity calculation is not affected by the size, this paper uses the cosine phase similarity as the benchmark to measure the similarity of the feature maps.

The quantity factor is as follows:

$$\text{Value}_i = \begin{cases} \frac{P_i^* \cdot P_k}{\|P_i^*\| \cdot \|P_k\|} & i \neq k \\ 1 & i = k \end{cases} \tag{2}$$

$$\varepsilon_i = \text{Soft max}\left(\text{Value}_i\right) = \frac{e^{\text{Value}_i}}{\sum_{j=2}^{6} e^{\text{Value}_i}} \tag{3}$$

The CNN method first converts the output feature map into 1×1 feature points through the multi-layer *Valid* convolution method. Then calculate the similarity between the feature value obtained in each layer and the feature value of the specified layer. Finally, the similarity result is used as a quantitative factor to determine the number of candidate regions in the corresponding feature layer. Among them, the specific layer is obtained from the baseline method. Since

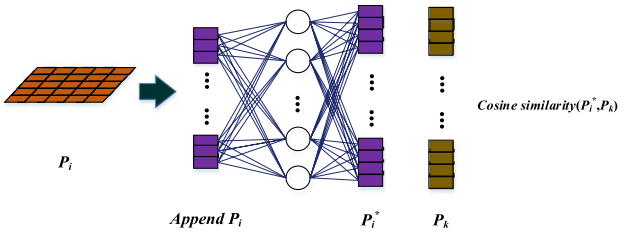

Fig. 4 Quality score module based on feedforward neural network

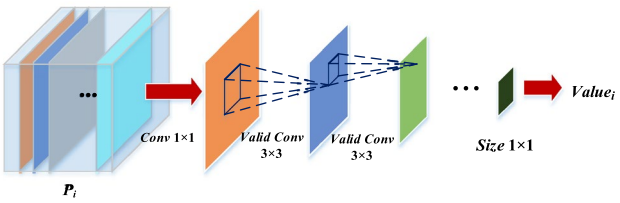

Fig. 5 Quality score module based on convolutional neural network

every output feature of FPN is derived from the input image, we think that there are similarities between all feature maps (Fig. 4). The network structure is shown in Fig. 5 below:

In Fig. 5, $Value_i$ is the feature map of the output P_i of the i_{th} layer of FPN, and the weighting calculation method is as follows:

$$w_i \frac{\text{value}_k - \left|\text{Value}_k - \text{Value}_i\right|}{\text{Value}_k} \tag{4}$$

where Value_k is the feature map of the feature map P_k of the K_{th} layer selected by the baseline method.

3.1.3 Aggregation module

The main function of this module is to generate candidate region groups according to the quantity factor of each scale feature map. Specifically, a series of candidate regions are generated for each feature map. We completely retain the candidate region of the feature layer of the baseline method, and retain part of the candidate region and the remaining feature layer. Among them, the number of reserved candidate regions is determined by the quantity factor, which can be expressed by formula (6).

$$\text{Num}_i = N_i \times \varepsilon_i \tag{5}$$

$$N_i = H_i \times W_i \times \text{anchors} \tag{6}$$

where N_i is the number of candidate regions generated by the P_i feature layer, and Num_i is the reserved number. The number of candidate regions generated by each feature layer is determined by the size of the feature map. H_i and W_i,

respectively, represent the height and width of the Pi feature layer. The *anchors* represent the number of anchor points generated by each feature point, and is usually set to 9.

3.2 Loss function

The loss function of object detection is mainly composed of two parts, classification loss and positioning loss, which can be described as:

$$L_{\text{Loss}} = \frac{1}{N}\sum_i L_{\text{cls}}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_{\text{loc}}}\sum_i t_i L_{\text{loc}}\left(g_i, g_i^*\right) \tag{7}$$

where i is the anchor index, p_i is the classification probability of the anchor i, p_i^* is the probability that the anchor i is the true label; g_i is the coordinate vector of the predicted bounding box, g_i^* is that of the ground truth coordinate vector; t_i represents the positive and negative sample type. t_i is 1 if the anchor is positive, and 0 if not. In order to train the detection network, we need positive samples and their ground truth. Calculate the degree of overlap between each candidate box and the ground truth bounding box. Candidate boxes are defined as positive samples if the overlap is greater than the threshold (0.5). Finally, the candidate frame with the largest overlap is selected as the object.

3.2.1 Area cross entropy loss function

For the classification loss L_{cls}, the multivariate cross entropy is usually used, and a negative log likelihood function is applied to all object classifications. The specific expression is as follows:

$$L\left(p_i, q_i\right) = -\sum_{j=1}^{c} q_{ij} \times \log\left(p_{ij}\right) \tag{8}$$

Among them, q_{ij} is a one-hot vector, which is defined as follows:

$$q_{ij} = \begin{cases} 1 & i_{\text{th}} \text{ sample category is } j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where p_{ij} represents the probability of i_{th} sample belongs to category j. When calculating category probability, the *Softmax* function is used.

It is not difficult to find that the weight of all samples in the cross entropy loss function is 1, which is equivalent to ignoring the size of the object. However, our attention to multi-scale objects is different in real scenes. A study has shown that when detecting multi-scale targets at the same time, more attention needs to be devoted to small target detection [39].

In order to deal with the difficulty of detecting small objects and low-resolution objects, this paper quote the area weight idea in [37] and design an area cross entropy loss function. Our expectation is to design a weight parameter that depends on the target size, which assigns different parameters to different objects. Using only width or height as a weighting factor is not the best choice, due to the existence of some large aspect ratio targets, such as buses and coach. Therefore, an area-based weight parameter m_i is proposed. Due to the large difference in the area of the proposal, we normalized its area to between 0 and 1, and designed a monotonically decreasing function on the area. In order to prevent the weight from being too small, the weighting factor m_i remains greater than 1 and less than 2. For the definition of m_i, we refer to the expression of the SoftMax function, and defined as follows:

$$m_i = 1 + e^{-s_i} \tag{10}$$

where s_i represents the area of the i_{th} prediction frame.

Figure 6 shows the image representation of the weight factor m_i. It can be seen that a larger weight can be obtained when the area of the prediction frame is relatively small. In contrast, when the area of the prediction frame is relatively large, a smaller weight can be obtained.

In summary, the area cross entropy loss function defined in this article is:

$$L_{\text{Area}}\left(p_i, q_i\right) = -\sum_{j=1}^{c}\left(1 + e^{-s_i}\right) \times q_{ij} \times \log\left(p_{ij}\right) \tag{11}$$

Compared with the author's global area weight in [37], the difference is that we only focus on the object classification process. Because our intuition is that the final regression process is based on classification.

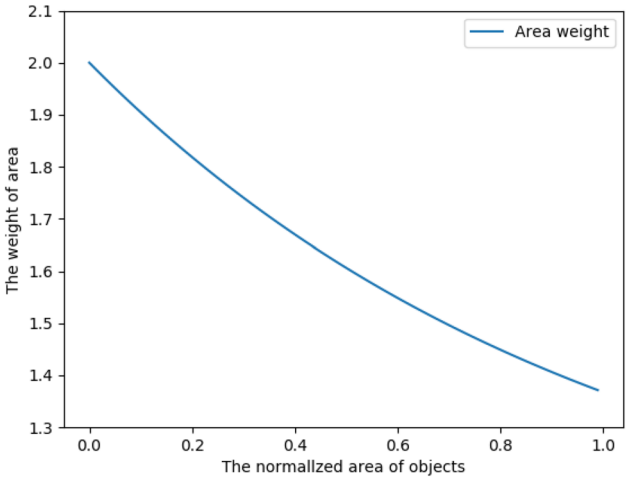

Fig. 6 Regional weight factor expression

3.2.2 Location loss

For the location loss, we choose the Smooth L_1 function with fast convergence speed and good smoothness, and its expression is:

$$\text{Smooth } L_1(x) = \begin{cases} 0.5x^2 & \text{if } x < 0 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{12}$$

In the overall loss calculation, $t_i \times L_{\text{loc}}$ means that only the positive sample loss will be activated. In the process of returning the candidate frame to the ground truth, the offset of the center (cx, cy), height (h) and width (w) can be expressed as:

$$\begin{aligned} \widehat{g}_j^{cx} &= \frac{\widehat{g}_j^{cx} - d_i^{cx}}{d_i^w}, \quad \widehat{g}_j^{cy} = \frac{\widehat{g}_j^{cy} - d_i^{cy}}{d_i^h} \\ \widehat{g}_j^{w} &= \log\left(\frac{g_j^w}{d_i^w}\right), \ \widehat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned} \tag{13}$$

In summary, our loss calculation can be defined as:

$$L_{\text{Loss}} = \frac{1}{N_{\text{cls}}} \sum_i (1 + e^{-s_i}) L_{\text{cls}}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{\text{loc}}} \sum_i t_i L_{\text{loc}}\left(g_i, g_i^*\right) \tag{14}$$

After adding the weight m_i to L_{Loss}, the size of the object will affect the loss and gradient, and the smaller the object, the greater the impact on the result.

4 Experiments

This section reports experimental details, including object detection data set, experimental environment, evaluation metrics, implementation details, experimental results.

4.1 Data set and evaluation metrics

4.1.1 Data set

With the development of the object detection field, many challenging data sets have been released for further research, such as PASCAL VOC, COCO, KITTI. In order to evaluate our proposal, experiments were carried out on the UA-DETRAC and KITTI datasets [24].

4.1.1.1 ***UA-DETRAC data set*** **[40]** It is a challenging multi-target detection benchmark in real scenes. The data set contains a series of video sequences under different scenes and was shot in 24 different locations in Beijing and Tianjin, China. There are more than 140,000 video frame pictures in the entire data set, including 84 K for the training set and 56 K for the test set. Since only the training data contains vehicle annotation information, we divide the training set into two parts, of which 56 K is used for training and 28 K is used for validation.

4.1.1.2 ***KITTI data set*** **[41]** It is the most representative object detection data benchmark in autonomous driving scenarios. Most of the pictures in KITTI are taken from the driving recorder and contain real picture data under various road conditions. This paper mainly selects the data set of the vehicle detection part, including 7 K training pictures and 7 K test pictures. Since only the training data contains vehicle annotation information, we also divide the training set into two parts, where 4 K is used for training and 3 K is used for validation.

4.1.2 Evaluation metrics

The COCO python package is used in ablation experiments, using AP_s, AP_m, and AP_l to verify the effectiveness of CRAN and area cross entropy. In addition, in order to verify the performance of the proposed method in the state-of-the-art network architecture, we also conducted experiments on the recent DETRAC benchmark [40] and KITTI benchmark [41].

4.2 Implementation details

4.2.1 Pre-processing

This article first selects ResNet50 as the backbone network, and introduces FPN to extract multi-scale feature map. The input size of DETRAC is 540×960 pixels, and the input size of KITTI is 576×1920 pixels. The generalization ability of the model has been improved by means of data set enhancement.

4.2.2 Training

In order to obtain a more accurate mapping, all our parameter settings follow the settings in [22]. The training set is used to train the network, and the validation set is used to verify the training results. In the training process, the batch size of each GPU is 4. In addition, the "Xavier" method in this paper is used to initialize the convolutional layer parameters, and the stochastic gradient descent (SGD) method is used to optimize the model. In particular, 12 epochs are set in the training process, the initial learning rate is 0.01, and it decreases to 50% of the current learning rate as the epoch increases, and the learning rate decays to 0.0001 after the 9th epoch. In addition, we use

optimization techniques such as batch normalization and dropout to optimize each method.

4.2.3 Testing

In the testing process, we use the trained object detection network model to obtain the category and border of each object in the test set, and then compare it with the label data to obtain its testing accuracy.

4.2.4 Experimental environment

Our experiment is based on python language and Pytorch1.2 framework in ubuntu16.04 operating system. The main hardware configuration includes 2.4 GHz CPU and 64 GB RAM. On this basis, GTX 1080Ti graphics cards (12G memory) are used for accelerated training.

4.3 Ablation analysis

This paper designs an ablation experiment on the COCO dataset to verify the performance of the proposed CRAN and area cross entropy on different evaluation indicators. During the experiment, the same parameter settings were used, and mAP with an IoU threshold of 0.7 was used to ensure the fairness of the experiment.

4.3.1 Baseline setting

In this paper, a baseline network based on Faster RCNN is constructed, the backbone network used is ResNet50, and FPN is used for multi-scale feature extraction. Table 1 shows the output feature size of FPN. Experimental results show that the global mAP is 36.5% in the test set [22].

Table 1 FPN output feature map size

Layer name	P2	P3	P4	P5	P6
Stride	64	32	16	8	4

4.3.2 Effect of CRAN module

CRAN module is applied to Faster R-CNN, and the basic network settings are consistent with the baseline method. The experimental results are shown in Table 2. It can be seen that the detection results have been improved after adding the CRAN module. During the experiment, we choose the CRAN module based on FNN and the CRAN module based on CNN to conduct experiments, respectively. From the experimental results in Table 2, it can be seen that the detection accuracy of the CNN-based method is better than FNN on the verification set, and the model size is also lower than the latter. This is understandable, because the FNN-based method contains more training parameters, and the CNN-based method has more advantages in processing two-dimensional data. Therefore, in the subsequent experiments, this article uses the CNN-based CRAN method.

Baseline method has a suboptimal performance on recalling objects of various scales, especially the tiny ones, as depicted in Fig. 7a. As shown in Fig. 7b, our CRAN performs considerably well, and achieves an encouraging recall rate on the COCO validation set.

In addition, this article also conducted experiments on the DETRAC dataset. The visualization results in Fig. 8 further illustrate the effectiveness of our mothed. Our CRAN performs considerably well and achieves an encouraging recall rate over 99% on the DETRAC validation set.

4.3.3 Effect of area cross entropy loss function

Three experiments are designed in this paper to verify the effectiveness of the area cross entropy loss function. The area cross entropy loss function is applied to RPN classification, object classification, and both simultaneously. Table 3 reports the comparison results between the three experimental methods and the baseline method. It can be seen that the proposed area cross entropy loss function has improved performance for the RPN classification process and the object classification process, especially for the detection of small targets, which also verifies our ideas. We found that when the area cross entropy loss function is applied to both RPN classification and object classification processes, the mAP has been greatly improved. Therefore, in the subsequent

Table 2 Ablation analysis on CRAN module

Method	mAP	AP_S	AP_M	AP_L	Training time	Model size	Environment
Baseline[25]	36.5	21.9	40.4	46.8	–	238 M	–
Random[22]	34.8	19.0	39.3	45.2	–	–	–
sum[22]	36.8	22.0	41.0	47.2	–	**330 M**	GPU@1080Ti
+CRAN(FNN)	36.9	22.0	40.8	**47.4**	73 h	289 M	GPU@1080Ti
+CRAN(CNN)	**37.3**	**22.3**	**41.5**	47.3	**52 h**	253 M	GPU@1080Ti

Boldface indicates the best performance among the comparison methods

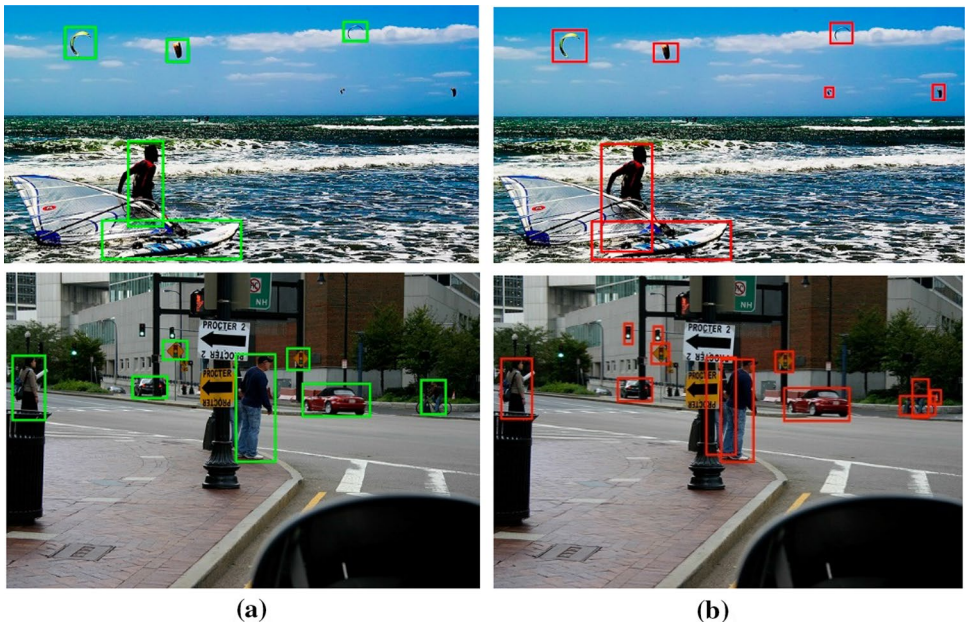

Fig. 7 Baseline **a** and CRAN **b**. Visibly, CRAN obtained better accuracy, and experimentally in COCO datasets, it has a higher recall for than baseline

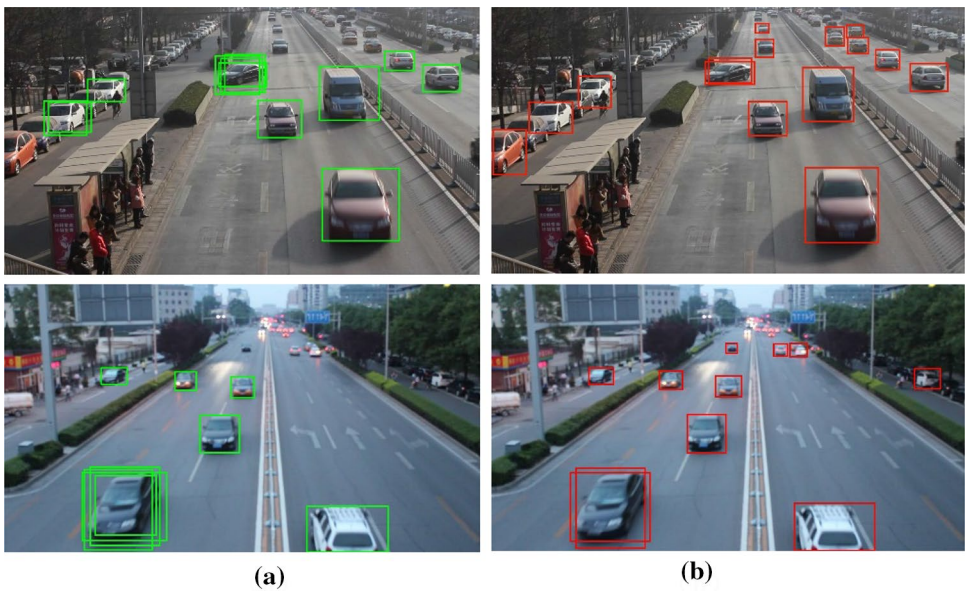

Fig. 8 Baseline **a** and CRAN **b**. Baseline **a** and CRAN **b**. Visibly, CRAN obtained better accuracy, and experimentally in DETRAC datasets, it has a higher recall for than baseline

Table 3 Ablation analysis on Area loss

Method	mAP	AP_S	AP_M	AP_L	Time
Baseline[25]	36.5	21.9	40.4	46.8	0.1 s/img
+ Area RPN_{cls}	37.0	**22.6**	40.6	46.8	0.1 s/img
+ Area CLS_{cls}	37.0	22.3	**40.8**	47.0	0.1 s/img
+ Area CLS	**37.2**	22.5	**40.8**	**47.2**	0.15 s/img

Boldface indicates the best performance among the comparison methods

experiments, this paper applies the area cross entropy loss function to both the RPN classification and object classification processes.

Figure 8 shows the loss in the training process. It can be seen from Fig. 9a that our area cross entropy loss can converge faster compared to the baseline method, and the overall loss value is smaller; Figure 9b shows the classification and positioning loss of our method separately. Obviously, the contribution of the classification loss to the overall loss is greater at the beginning of training, which also validates the idea in Sect. 3.2 of this paper.

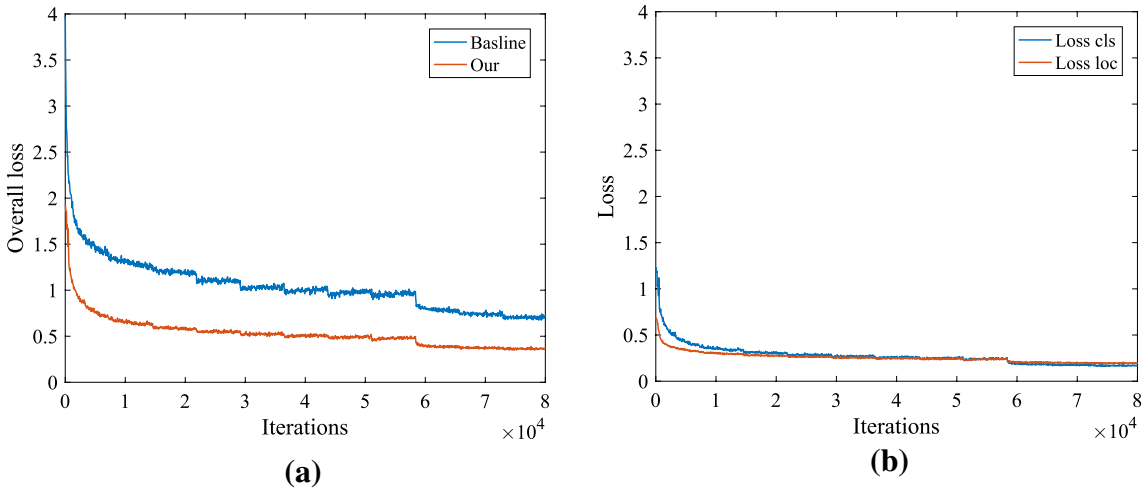

Fig. 9 Loss during training. **a** baseline mothed overall loss and our overall loss. Area cross entropy loss can converge faster and the loss value is smaller. **b** Our classification and positioning loss

4.4 Application of CRAN and area cross entropy to different architectures

We apply the proposed CRAN and area cross entropy to the several state-of-the-art architectures at present, and verify the performance of the method in the UA-DETRA dataset and the KITTI dataset. For the one-stage network, our method eliminates the RPN process. CRAN is applied to aggregate candidate regions of different feature layers, and the area cross entropy is only applied to the object classification process.

4.4.1 Performance test on UA-DETRAC data set

This paper tests our method on the UA-DETRAC dataset, and submitted the training results of the training set and validation set to the UA-DETRAC benchmark test. The comparative experimental results are reported in Table 4. It can be seen that our proposed method performs well in outstanding detectors. In particular, the performance on the Hard subset has been greatly improved, which is consistent with our original intention of designing area cross entropy. It is worth noting that our proposed method has a good performance on the two-stage detector, which improves the detection accuracy by more than 1.0% on average. In addition, the test results of several outstanding anchor-free methods on UETRAC (CornerNet, CenterNet, FCOS) are also listed. We found that the method proposed in this article makes some excellent detectors beyond the anchor-free method.

4.4.2 Performance detection on KITTI data set

In order to verify the performance of the proposed method in state-of-the-art network structures, we also conducted training and testing on the KITTI dataset, and fully evaluated our method on the KITTI benchmark. Applying our

Table 4 Performance evaluation on UA-DETRAC dataset. + + means adding our proposed CRAN module and area cross entropy loss; (+**) means improved detection performance

Method	Overall	Easy	Medium	Hard	Sunny	Cloudy	Rainy	Night	Time
Faster RCNN + +	**60.06(+1.61)**	83.59	63.96	45.77	63.15	66.97	45.86	71.13	0.06 s/img
FCN + +	**71.25(+1.38)**	93.87	76.42	55.69	85.22	74.94	57.34	76.57	0.09 s/img
EB + +	**69.25(+1.29)**	89.82	74.57	56.03	73.26	74.85	54.07	84.65	0.06 s/img
Yolo v2 + +	**58.33(+0.61)**	83.48	63.86	44.16	70.47	57.78	48.36	65.49	0.04 s/img
Yolo v3 + +	**73.29(+0.83)**	95.26	77.32	52.67	87.64	78.36	57.15	79.29	0.05 s/img
RetinaNet + +	**78.27(+0.86)**	95.85	81.14	61.35	88.39	80.23	59.22	82.41	0.06 s/img
Cascade R-CNN + +	**79.34(+1.26)**	95.32	83.69	64.28	87.34	79.59	60.37	84.17	0.08 s/img
CornerNet	76.64	92.86	81.65	59.66	85.34	77.41	58.20	78.79	–
CenterNet	78.22	94.49	83.22	61.91	87.47	80.74	57.49	81.63	–
FCOS	79.15	95.07	83.65	63.47	88.40	80.67	61.39	86.33	–
SpineNet + +	**82.16(+1.34)**	96.34	83.39	64.19	89.44	82.61	62.88	87.84	0.08 s/img
CBNet + +	**83.29(+0.92)**	96.81	85.12	66.28	90.37	83.94	64.31	89.25	0.08 s/img

Boldface indicates the best performance among the comparison methods

Table 5 Performance evaluation on KITTI dataset. + + means adding our proposed CRAN module and area cross entropy loss; (+**) means improved detection performance

Method	Average Precision (AP)/%			Time
	Easy	Moderate	Hard	
Faster RCNN + +	88.65(+0.75)	79.57(+0.46)	71.46(+1.27)	1.4 s/img
RefineNet + +	90.26(+0.10)	79.82(+0.61)	66.29(+0.58)	1.4 s/img
MSCNN + +	90.58(+0.12)	89.19(+0.36)	**77.25(+2.41)**	0.2 s/img
YOLO v2 + +	**89.53(+1.52)**	86.44(+0.79)	76.23(+2.07)	0.03 s/img
Yolo v3 + +	93.31(+0.91)	89.72(+1.14)	77.94(+1.88)	0.05 s/img
RetinaNet + +	94.67(+0.94)	90.34(+0.76)	78.62(+1.64)	–
Cascade R-CNN + +	94.86(+1.03)	90.92(+0.83)	79.92(+1.51)	–
CornerNet	92.80	87.54	76.52	0.08 s/img
CenterNet	93.42	87.81	76.83	0.08 s/img
FCOS	94.26	89.20	78.09	–
SpineNet + +	95.28(+0.85)	**90.73(+1.23)**	79.47(+1.32)	–
CBNet + +	95.69(+0.79)	91.21(+1.19)	81.04(+1.17)	1.1 s/img

Boldface indicates the best performance among the comparison methods

Fig. 10 Success cases from DETRAC and KITTI

method to several outstanding detectors, Table 5 gives a comparison of several methods. It can be clearly seen that for several outstanding detectors, the proposed method improves the mAP by more than 1.0%, especially on the Hard subset.

This improvement is more obvious in Fig. 10, where we visualize some detection cases on the DETRAC and KITTI test sets. It can be clearly seen from the successful cases that the proposed method can better detect small targets at a longer distance. In particular, it can also detect obstructed, blurred, and night vehicles. In short, our method can not only be applied to a variety of detectors, but also enhance the generalization ability of the network.

5 Conclusion

In order to improve the performance of vehicle object detection, this paper proposed a CRAN and area cross entropy loss, respectively, to improve the recall rate of the model and the detection performance of difficult instances. The ablation experiment proves that the proposed method can not only greatly improve the recall rate, but also promote model convergence. Finally, the experimental results on the UA-DETRAC and KITTI datasets show that our method can increase the mAP of several existing advanced detectors by more than 1%, especially the two-stage detector.

Acknowledgements This work was supported by the Nondestructive Detection and Monitoring Technology for High Speed Transportation Facilities, Key Laboratory of Ministry of Industry and Information Technology, and the Fundamental Research Funds for the Central Universities, NO.NJ2020014.

References

1. Tian Y, Du Y, Zhang Q, et al. (2020) Depth estimation for advancing intelligent transport systems based on self-improving pyramid stereo network. Inst Eng Technol 14(5):338–345. https://doi.org/10.1049/iet-its.2019.0462
2. Liu W, Liao S, Hu W (2019) Towards accurate tiny vehicle detection in complex scenes. Neurocomputing 347:24–33
3. Girshick R, Donahue J, DaT Tell T, Malik J. Rich feature hierarchies for accurate object detection and Semantic segmentation //Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE: 580–587[DOI:10.1109/ CVPR.2014.81]
4. Girshick R, landola F, Darrell T, Malik J. Deformahle part models are convolutional neural networks// Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE: 437–446 [DOI: 10. 1109/ CVPR.2015.7298641]
5. Ren S, He K, Girshick R et al (2015) Faster R-CNN: towards real-time object detection with region proposal networks//. Adv Neural Inf Process Syst, IEEE. https://doi.org/10.1109/TPAMI.2016.2577031
6. Dai J, Li Y, He K, Sun J. R-FCN: Object detection via region-based fully convolutional networks. arXiv preprint https://arXiv.org/1605.06409, 2016
7. Uijlings JRR, van de Sande KEA, Gevers T et al (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171. https://doi.org/10.1007/s11263-013-0620-5
8. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings IEEE Conference Comput. Vis. Pattern Recognit. (CVPR), pp 779–788
9. Liu W et al. (2016) SSD: single shot MultiBox detector. In: Proceedings Eur. Conf. Comput. Vis. pp 21–37
10. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings IEEE Int. Conf. Comput. Vis. pp 2980–2988
11. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement", arXiv: 180402767 Cs
12. Fu CY, Liu W, Ranga A et al. (2017) "DSSD: Deconvolutional single shot detector,". [Online]. Available: https://arxiv.org/1701.06659
13. He K, Gkioxari G, Doll´ar P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
14. Lin TY, Doll´ar P, Girshick R, et al. (2017) Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp2117–2125
15. Rossi L, Karimi A, Prati A (2021)A novel region of interest extraction layer for instance segmentation. Comput Vis Pat Recog. https://arxiv.org/2004.13665v2
16. Farahani G (2017) Dynamic and robust method for detection and locating vehicles in the video images sequences with use of image processing algorithm[J]. Springer International Publishing,(1)
17. Lienhart R, Maydt J (2002) An extended set of Haar-like features for rapid object detection[C]//International Conference on Image Processing, pp 900–903
18. Lowe DG (2004) Distinctive image features from scale-invariant keypoints[J]. Int J Comput Vis 60(2):91–110
19. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection[C]//. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp 886–893
20. Felzenszwalb PF, Girshick RB, McAllester D et al (2009) Object detection with discriminatively trained part-based models[J]. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
21. Hong-Peng YIN, Bo CHEN, Yi CHAI et al (2016) Vision-based object detection and tracking: a review [J]. Acta Autom Sin 42(10):1466–1489
22. Ciresan DC, Meier U, Masci J, et al. (2011) High-performance neural networks for visual object classification [J]. arXiv: 1102.0183
23. Everingham M, Eslami SMA, Van Gool L et al (2014) The PASCAL visual object classes challenge: a retrospective. Int J Comput Vis 111:98–136 (2015). https://doi.org/10.1007/s11263-014-0733-5
24. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
25. Cai Z, Vasconcelos N (2019) Cascade r-cnn: high quality object detection and instance segmentation. In: IEEE transactions on pattern analysis and machine intelligence, pp 1483–1498. https://doi.org/10.1109/TPAMI.2019.2956516
26. Kong T, Yao A, Chen Y, Sun F (2016) Hypernet: towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp 845–853
27. Lee H, Eum S, Kwon H (2017) In: ME R-CNN: multi-expert region-based CNN for object detection. https://arxiv.org/abs/1704.01069v1
28. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp 761–769
29. Wang X, Shrivastava A, Gupta A (2017) A-fast-RCNN: hard positive generation via adversary for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, pp 3039–3048
30. Bell S, Lawrence Zitnick C, Bala K, Girshick R (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2874–2883
31. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic seg- mentation-aware CNN model. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1134–1142
32. Shrivastava A, Gupta A (2016) Contextual priming and feedback for faster R-CNN. In: Proceedings of the European Conference on Computer Vision, Springer, pp 330–348
33. Redmon J, Farhadi A (2017) YOLO9000: Better, faster, stronger. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog, pp 6517–6525
34. Lee W-J, Kim DW, Kang T-K, Lim M-T (2018) Convolution neural network with selective multi-stage feature fusion: case study on vehicle rear detection. Appl Sci 8:2468. https://doi.org/10.3390/app8122468
35. Pae DS, Choi IH, Kang TK et al (2018) Vehicle detection framework for challenging lighting driving environment based on feature fusion method using adaptive neuro-fuzzy inference system. Int J Adv Robot Syst. https://doi.org/10.1177/1729881418770545
36. Guo Y, Xu Y, Li S (2020) Dense construction vehicle detection based on orientation-aware feature fusion convolutional neural network[J]. Autom Constr 112
37. Wang P, Sun X, Diao W, Fu K (2020) FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale

remote sensing imagery. IEEE Trans Geosci Remote Sens 58(5):3377–3390
38. Gu Y, Wang B, Xu B (2018) A FPN-based framework for vehicle detection in aerial images. In: ICVIP 2018: Proceedings of the 2018 the 2nd international conference on video and image processing, pp 60–64. https://doi.org/10.1145/3301506.3301531
39. Weymar M, LW A, Hman A, et al (2011) The face is more than its parts--brain dynamics of enhanced spatial attention to schematic threat. Neuroimage 58(3):946-954
40. Wen L, Du D, Cai Z, et al. (2015) UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking
41. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The kitti vision benchmark suite.In: Proceedings of the 2012 IEEE Conference on Com- puter Vision and Pattern Recognition (CVPR),,IEEE, pp 3354–3361

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.