Hindawi
Complexity
Volume 2021, Article ID 5555121, 10 pages
https://doi.org/10.1155/2021/5555121

Research Article

Multiscale Feature Learning Based on Enhanced Feature Pyramid for Vehicle Detection

Hoanh Nguyen

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh, Vietnam

Correspondence should be addressed to Hoanh Nguyen; nguyenhoanh@iuh.edu.vn

Received 21 April 2021; Revised 16 May 2021; Accepted 4 June 2021; Published 14 June 2021

Academic Editor: Kai Hu

Vehicle detection is a crucial task in autonomous driving systems. Due to large variance of scales and heavy occlusion of vehicle in an image, this task is still a challenging problem. Recent vehicle detection methods typically exploit feature pyramid to detect vehicles at different scales. However, the drawbacks in the design prevent the multiscale features from being completely exploited. This paper introduces a feature pyramid architecture to address this problem. In the proposed architecture, an improving region proposal network is designed to generate intermediate feature maps which are then used to add more discriminative representations to feature maps generated by the backbone network, as well as improving the computational cost of the network. To generate more discriminative feature representations, this paper introduces multilayer enhancement module to reweight feature representations of feature maps generated by the backbone network to increase the discrimination of foreground objects and background regions in each feature map. In addition, an adaptive RoI pooling module is proposed to pool features from all pyramid levels for each proposal and fuse them for the detection network. Experimental results on the KITTI vehicle detection benchmark and the PASCAL VOC 2007 car dataset show that the proposed approach obtains better detection performance compared with recent methods on vehicle detection.

1. Introduction

Vehicle detection is an important task in autonomous driving systems, traffic control, management, and so on. With the fast development of autonomous driving in recent years, vehicle detection methods based on computer vision are getting more attention. Traditional approaches for detecting vehicles in images usually use motion and handcrafted features of vehicles such as colour, edge, character, and texture to locate their position. However, with a variety of vehicle orientations and scales in urban driving, along with occlusions, detecting vehicles based on handcrafted features is a challenging problem.

In general, vehicle detection can be considered as a special topic of generic object detection. In recent years, deep convolutional neural network (CNN) object detectors such as faster R-CNN [1], SSD [2], and YOLOv2 [3] have achieved significant improvements on general object detection compared with conventional methods. However, these object detectors are based on single scale feature map for detecting objects, thus limiting the detection performance of detecting multiscale objects and objects in difficult conditions. To further improve the detection performance, some frameworks such as FPN [4] and RetinaNet [5] proposed using multiscale feature maps generated by backbone network for locating objects. The idea of using multifeature representations and hybrid fusion has been proposed in many fields [6] and achieved certain successes. However, using multiscale feature map generated by the base network with less semantic information prevents the multiscale features from being completely exploited [7].

Motivated by the above research ideas, this paper proposes a deep CNN approach based on faster R-CNN with FPN backbone for vehicle detection. In the proposed framework, an improving region proposal network (RPN) is introduced to generate intermediate feature maps which are then used to generate enhanced feature maps with more discriminative representations by a multilayer enhancement

module. The multilayer enhancement module contains only simple operations to reduce computational cost. In addition, an adaptive RoI pooling module is proposed to pool features from all pyramid levels for each proposal and fuse them for the detection network. In addition, an adaptive RoI pooling layer is designed to pool features from all pyramid levels and fuse them for the detection network.

The main contributions of this paper can be summarized as follows:

(i) A novel multilayer enhancement module is introduced to generate more discriminative representation feature maps. By improving multiscale features with strong semantics, the performance of vehicle detection has been significantly improved.

(ii) For generating proposals, an adaptive RoI pooling module is designed to better exploit RoI features from different pyramid levels and produce a better RoI feature for subsequent location refinement and classification.

(iii) Using a two-stage strategy with enhanced feature maps and multiscale feature learning, the proposed approach achieves better detection accuracy than other state-of-the-art methods on vehicle detection.

The remaining parts of this study are organized as follows. Section 2 reviews recent works on vehicle detection. Section 3 elaborates the proposed approach. Section 4 presents the experimental results and comparison between the proposed framework and recent frameworks. Finally, the conclusions are reviewed in Section 5.

2. Related Work

This section gives a brief introduction of recent deep CNN detection frameworks based on multilayer features and vehicle detection frameworks based on computer vision, including traditional methods and deep CNN methods.

2.1. Deep CNN Detection Frameworks Based on Multilayer Features. Deep CNN detection framework such as faster R-CNN or SSD have achieved significant performance on object detection compared with conventional methods. However, these popular object detectors are based on single scale feature map for detecting objects, thus limiting the detection performance of detecting multiscale objects and objects in difficult conditions. Some frameworks such as FPN [4] and RetinaNet [5] proposed to use multiscale feature maps generated by backbone network at different layers for locating objects. To further improve the detection performance of detecting objects in difficult conditions based multiscale features, multilayer proposal [8] applied a deconvolution module to a lightweight architecture to generate enhanced feature map which can improve small object detection. MFANet [9] proposed a multilevel feature aggregation network which first extracts the deep features and filters the redundant channel information to optimize the learned context and then uses high-level features to provide guidance information for low-level features. In DAU-Net [10], deep feature information is fused with shallow feature information through multiscale attention modules to improve to the accuracy of water segmentation. In [11], the authors introduced an improved atrous spatial pyramid pooling method to extract the multiscale deep semantic information. The global attention up-sample mechanism was used to fuse deep semantic information with shallow spatial information, which improved ability to utilize global and local features.

2.2. Vehicle Detection. A vehicle detection system based on computer vision typically consists of two parts: locating vehicle candidate regions and classifying candidate regions. Conventional methods for vehicle detection are usually based on motion and appearance of vehicles in images to locate them. Motion-based methods use the motion to detect the vehicles in image frame. Background subtraction [12] and optical flow [13] are most widely used methods that are based on the motion of vehicles. To improve detection performance, optical flow is combined with symmetry tracking [14] and handcrafted appearance features [15]. Appearance-based methods usually adopt external physical features of vehicles such as colour, texture, edge, and shape to locate vehicles. These methods first define regions in image by some feature descriptors and then classify these regions into different classes such as vehicle and background. Variety feature descriptors have been used in this field such as HOG [16], SURF [17], Gabor [18], and Haar-like [19]. These feature descriptors are usually followed by classifiers such as SVM [16], artificial neural network [18], and Adaboost [19]. Conventional methods have achieved a certain level of success in vehicle detection. However, with a variety of vehicle orientations and scales in urban driving, along with occlusions, detecting vehicles based on handcrafted features is still a challenging problem.

In recent years, deep CNNs have demonstrated superior performance in various tasks compared to conventional learning methods. As a result, many vehicle detection methods based on deep CNN have been proposed and achieved significant improvements. In [20], the authors introduced a vehicle detection approach based on multitask deep CNN, including category classification, bounding box regression, overlap prediction, and subcategory classification. A region-of-interest voting scheme and multilevel localization were also introduced to further improve detection accuracy and reliability. This approach achieved satisfactory results on standard dataset. In [21], the deep model was used for vehicle detection that consists of feature extraction, deformation processing, occlusion processing, and classifier training using the back-propagation algorithm to enhance the potential synergistic interaction between various parts and to get more comprehensive vehicle characteristics. Li et al. [22] introduced a deep network which consists of YOLO under the Darknet framework for multivehicle detection from traffic video. The detection results showed that the method was suitable for the multiple target detection of different traffic densities. Wang et al. [23] proposed a real-time vehicle detection algorithm which fuses

vision and lidar point cloud information to take the advantages of the depth information generated by lidar sensor and the obstacle classification ability generated by vision sensor. The proposed algorithm significantly improved vehicle detection accuracy, especially for vehicles with severe occlusion. The authors in [24] introduced a flexible bounding-box generating algorithm to locate vehicle candidate regions and a graph-based algorithm to compute a vehicle proposal score for each bounding box. With these algorithms, the vehicle detection problems such as false alarms and occlusions in complex backgrounds were solved effectively. In [25], a scale-insensitive CNN is introduced to detect vehicles with a large variance of scales accurately and efficiently. The scale-insensitive CNN included context-aware RoI pooling scheme to preserve the original structures of small-scale objects and multibranched decision network which each branch is designed to minimize the intraclass distance of features to effectively capture the discriminative features of objects with various scales. In [26], the author proposed an improved framework based on faster R-CNN for fast vehicle detection. The proposed framework adopted MobileNets architecture [27] to build the base convolution layer in Faster R–CNN. In addition, Soft-NMS algorithm was used to solve the issue of duplicate proposals and context-aware RoI pooling layer was adopted to adjust the size of proposals without sacrificing important contextual information.

3. Methodology

This section presents the details of the proposed deep CNN-based framework for vehicle detection. The proposed framework is based on the faster R-CNN with FPN backbone [4].

3.1. Improving RPN. The faster R-CNN with FPN backbone [4] adopts the RPN at different feature maps for generating proposals with different scales and aspect ratios. The original RPN includes a $3 \times 3 \times 256$ convolution layer followed by two parallel 1×1 convolution layers for classifying and regressing proposals. Since the RPN is trained to locate foreground regions under the supervision of ground truth regions, the intermediate feature layer in the RPN contains discriminative features between foreground regions and background regions and can be used to enhance other feature layers by strengthening foreground features and suppressing background features. The more discriminative information in the intermediate feature layer generated by the RPN, the more discriminations of foreground objects and background regions are produced in the enhanced feature layers generated from the intermediate feature layer, which can improve the overall detection performance. Based on this analysis, this paper designs an improving RPN based on depth-wise dilated separable convolutions [28] to not only facilitate the intermediate feature layer in the RPN but also reduce computational cost. Depth-wise dilated separable convolutions use lightweight filters by factoring a standard convolution into two layers: depth-wise dilated convolution with a dilation rate of r to learn representations from large effective receptive field and point-wise convolution to learn linear combinations of input. Depth-wise dilated separable convolutions also provide better computational cost compared to standard convolution layers.

The structure of the proposed RPN is shown in Figure 1. Based on the structure of depth-wise dilated separable convolutions, this paper first replaces $3 \times 3 \times 256$ convolution layer in the original RPN by a 3×3 depth-wise dilated convolution layer with dilation rate of 2. This results in a larger receptive field. Large receptive field will encode more discriminative information in the intermediate feature layer generated by the RPN, which facilitates the enhanced feature map generated by the multilayer enhancement module as described in Section 3.2. Then, a 1×1 convolution layer with 256 channels is added to compares the number of channels of the output feature map. In practice, 3×3 depth-wise dilated convolution followed by $1 \times 1 \times 256$ convolution layer provides better runtime speed compared to $3 \times 3 \times 256$ counterparts while effectively enlarging the receptive field.

3.2. Multilayer Enhancement Module. High-resolution shallow features generated by deep CNNs network usually contain more detailed information such as edges and textures, while low-resolution deep features focus on semantics information of objects. Deep CNNs-based object detectors based on the last feature layer of the backbone network such as faster R-CNN are prone to miss detection for small objects. On the other hand, simple prediction based on multilayer features such as SSD can accelerate the inference speed of the network, but excessive use of shallow information makes the classification subnet less robust. In recent years, feature pyramids network FPN [4] has been widely applied in multiscale object detection. FPN fuses features of different scales by upsampling and summation in a top-down path. However, features at different scales generated by deep CNNs backbone usually contain information at different abstract levels, fusing multiple features with large semantic gaps between them would lead to a suboptimal feature pyramid. Normally, the feature representations of foreground objects need to be higher than that of background regions to achieve the best detection performance. To improve the detection performance based on this idea, the authors in [29] proposed PANet with a bottom-up path augmentation module to enhance the entire feature hierarchy with accurate localization signals in lower layers. However, due to different contributions of feature layers at different resolutions to augmentation path, treating feature layers of different importance equally will have a negative impact on the network [30]. Moreover, PANet increases the model parameters significantly. Motivated by these insights and the design of improving RPN in Section 3.1, this paper proposes a novel multilayer enhancement module based on the intermediate feature layers generated by the improving RPN to enhance multilayer feature maps generated by backbone network and improve the performance of detection network. The multilayer enhancement module reweights feature representations of feature maps generated

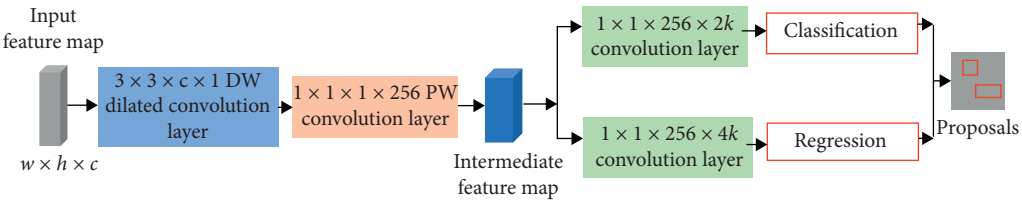

Figure 1: The structure of the improving RPN, each convolution layer is shown as kernel_size × number_of_kernels×number_of_output_channels.

by backbone network to increase the discrimination of foreground objects and background regions in each feature map. Figure 2 illustrates the structure of the multilayer enhancement module. First, the intermediate feature map from the improving RPN, which is trained to contain discriminative representations of foreground objects and background regions, is fed into a sigmoid layer after a batch norm layer to rescale the values of the output features within [0, 1]. Then, the feature map generated by the backbone is reweighted by multiplication operation with the corresponding output feature map. By rescaling, the values of the output features within [0, 1] and applying multiplication operation, the feature distribution in final feature map is refined by strengthening foreground features and suppressing background features. As a result, the final enhanced feature map contains more discriminative representations of foreground objects and background regions, which can improve the detection performance of the detection network. It should be noted that the multilayer enhancement module contains only simple operations; thus the computational cost of the module is insignificant. Figure 3 shows visualization of one feature map layer generated by backbone network (middle column) and that layer after enhancing by the multilayer enhancement module (right column). It is clear that the multilayer enhancement module effectively refines the feature distribution with foreground features enhanced and background features weakened.

3.3. Adaptive RoI Pooling Module. In the faster R-CNN with FPN backbone, proposals are pooled from one certain pyramid level, which is chosen according to the size of proposals, by a RoI pooling layer. In general, small proposals are extracted from low-level features and large proposals are extracted from high-level features. In some cases, two proposals with similar sizes may be extracted from different feature levels, which may produce nonoptimal results. To address this problem, PANet [29] introduced Adaptive Feature Pooling (AFP) structure, which pools features from all levels for each proposal and fuses them for following prediction. To be more specific, AFP structure first pools features from all levels by RoIAlign [31]. Then a fusion operation (element-wise max or sum) is used to fuse features from different levels after adapting them with fully connected layers. AFP structure improves the performance of instance segmentation. However, the extra fully connected layers in this structure increase the parameters significantly.

Motivated by AFP structure, this paper introduces an adaptive RoI pooling module to better exploit RoI features from different pyramid levels and produce better RoI features for subsequent location refinement and classification of vehicles, especially for small vehicles. Figure 4 shows the structure of the proposed adaptive RoI pooling module, which is simple in implementation. First, each RoI generated by the improving RPN subnet is mapped to all enhanced feature levels generated by the multilayer enhancement module. Then, context-aware RoI pooling (CARoI pooling) [25] is adopted to pool features from each level. CARoI pooling includes two operations to generate fixed size proposals: max pooling to reduce the size of proposals and deconvolution operation to enlarge the size of proposals. CARoI pooling provides better performance of detecting small objects compared to traditional RoI pooling and RoIAlign used in PANet [25, 26]; thus it can be used to enhance the performance of detecting small vehicles. Finally, instead of adapting the RoI features with fully connected layers like AFP structure, this paper uses max operation to fuse features from different levels. Since the proposed adaptive RoI pooling module does not introduce extra fully connected layers to adapt the RoI features, it would significantly decrease the model parameters. As a result, the proposed pooling module requires extremely fewer parameters compared to AFP structure in PANet while achieving comparable detection performance.

3.4. Network Architecture. The proposed network is built by using FPN backbone, multilayer enhancement module, improving RPN, and adaptive RoI pooling module as shown in Figure 5. Following FPN [4], this paper also adopts ResNet [32] as the basic network and uses {$P2$, $P3$, $P4$, $P5$} produced by feature pyramid as inputs of the improving RPN and the multilayer enhancement module. The improving RPN is applied to all different scale levels of feature maps {$P2$, $P3$, $P4$, $P5$} to generate corresponding intermediate feature layers and RoIs. As in [4], this paper also assigns anchors of a single scale and multiple aspect ratios at each level. To be more specific, this paper defines the anchors to have areas of $\{32 \times 32, 64 \times 64, 128 \times 28, 256 \times 256\}$ pixels on {$P2$, $P3$, $P4$, $P5$}, respectively. For trade-off between recall and inference speed, three anchor box ratios of $\{1:1, 1:2, 2:1\}$ are used at each location. Thus, there are total 12 anchors over the feature pyramid. Since there are many RoIs heavily overlapping with each other, nonmaximum suppression (NMS) algorithm is adopted to filter the number of RoIs before feeding them into the adaptive RoI pooling layer. This paper sets the intersection-over-union (IoU) threshold at 0.7 for NMS. Then, this paper assigns anchors training labels based

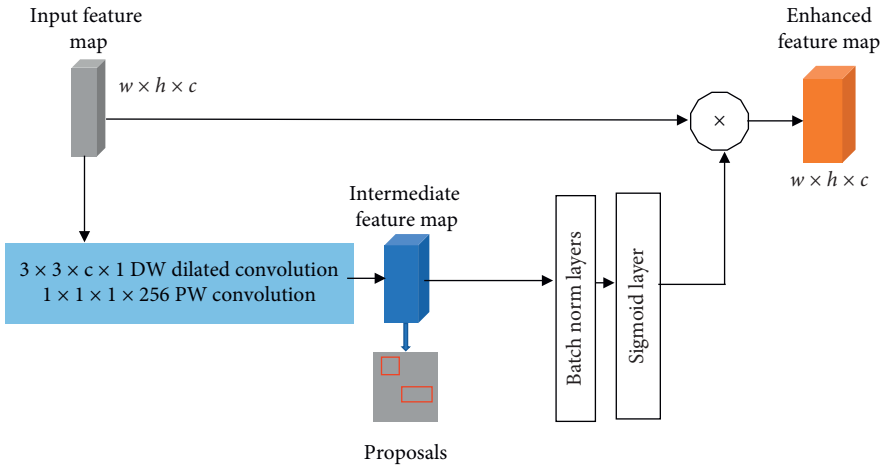

Figure 2: The structure of the multilayer enhancement module. Here, $c = 256$ channels for all feature maps generated by FPN backbone {$P2$, $P3$, $P44$, $P5$}.

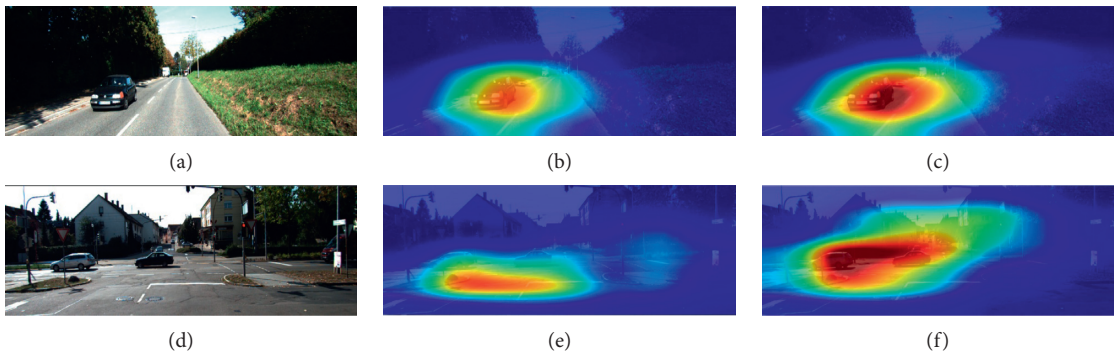

Figure 3: Visualization of one feature map layer generated by backbone network (middle column) and that layer after enhancing by the multilayer enhancement module (right column).

on their IoU ratios with ground truth bounding boxes. To be more specific, if the anchor has IoU over 0.7 with any ground truth box, it will be set as positive anchor. In addition, anchors which have the highest IoU for each ground truth box will also be assigned as positive anchor. Otherwise, if anchors have IoU less than 0.3 with all ground truth boxes, they will be set as negative anchor. The parameters of the improving RPN are shared across all feature pyramid levels. The multilayer enhancement module takes feature maps {$P2$, $P3$, $P4$, $P5$} generated by feature pyramid and intermediate feature layers produced by the improving RPN as inputs and generates corresponding enhanced feature maps {E2, E3, E4, E5}. These enhanced feature maps are then fed into the adaptive RoI pooling layer. Based on enhanced feature maps and RoIs produced by the improving RPN, the adaptive RoI pooling module generate a fixed size proposal corresponding with each RoIs. These fixed size proposals are fed into the R-CNN subnet for final prediction.

4. Experiments

In this section, this paper evaluates the proposed method on two public datasets: the KITTI benchmark [33] and the PASCAL VOC 2007 car dataset [34]. This paper compares the proposed method with recent methods on the KITTI testing set and the PASCAL VOC 2007 car dataset, and comprehensive ablation studies are conducted on the KITTI validation set. All experiments are supported by a Linux system with Intel Core I7-10700 CPU, 16 GB of memory, and Nvidia RTX 3070 GPU.

4.1. Dataset and Evaluation Metrics. The KITTI vehicle detection benchmark is a large automatic driving dataset. This dataset contains 7481 training images with available ground-truth labels and 7518 testing images without ground-truth labels. The main difficulty of car detection on the KITTI dataset is that a large number of cars are in small size (height < 40 pixels) and occluded. Since ground-truth labels of the KITTI test set are not available, this paper splits the KITTI training images into training set with 3682 images for training and validation set with 3799 images for validation as in [26, 35]. The validation set is used to conduct experiments to investigate the effectiveness of each module structure. Detection results are evaluated at three difficulty levels: easy, moderate, and hard as suggested by the KITTI

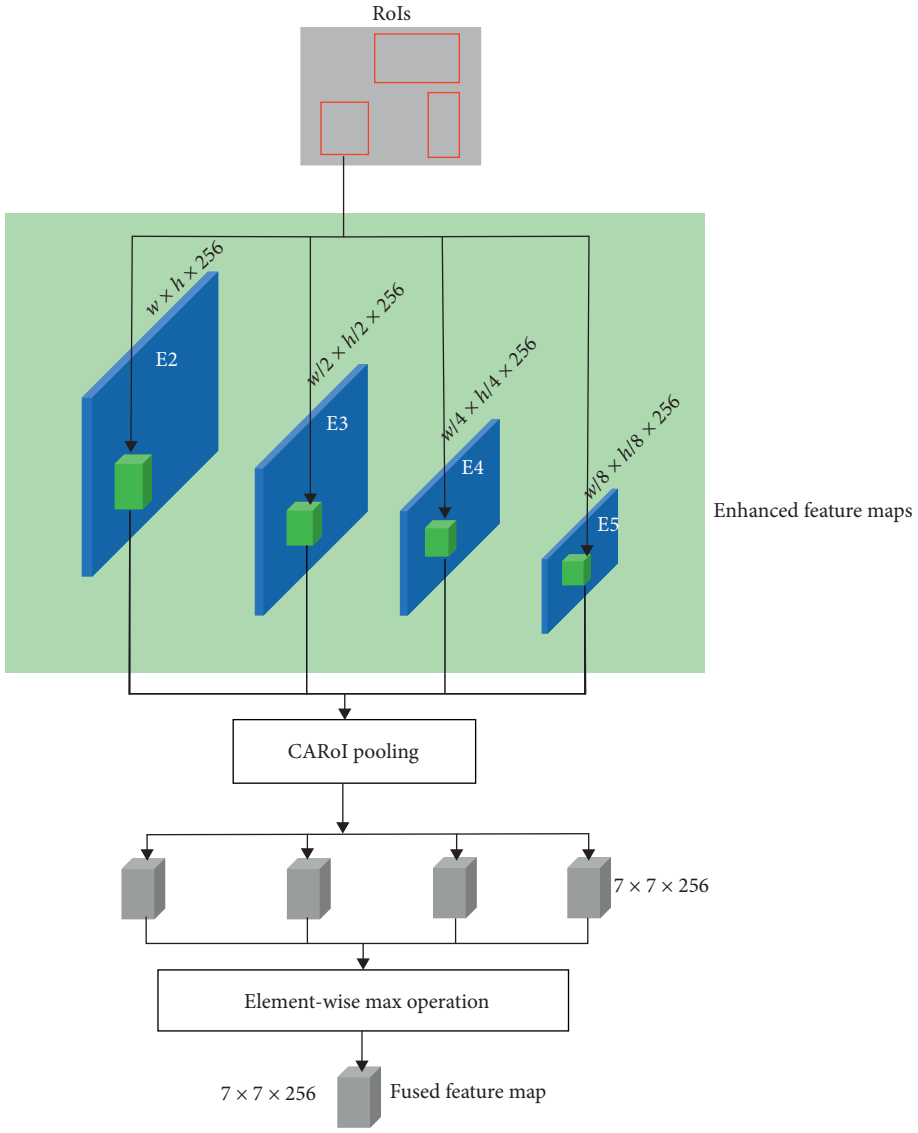

Figure 4: The structure of the adaptive RoI pooling module.

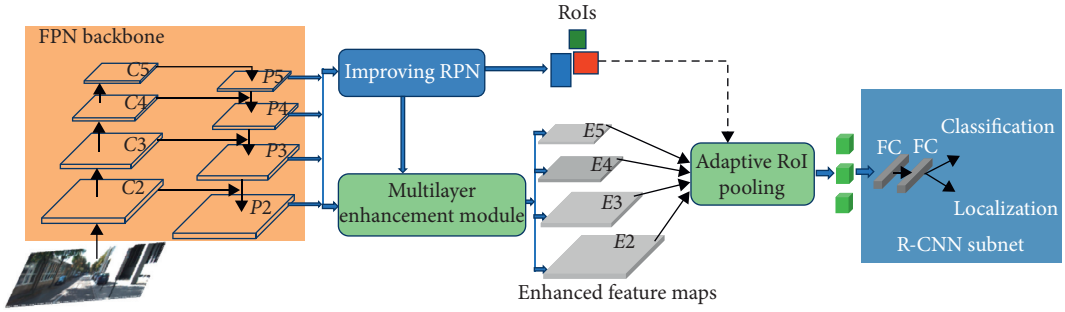

Figure 5: Network architecture.

benchmark [33]. To evaluate the object detection accuracy, the average precision (AP) is reported throughout the experiments, and 0.7 overlap threshold is adopted in the KITTI benchmark for car class.

The PASCAL VOC 2007 database is also used to evaluate the proposed framework. This dataset is composed of 5011 training images and 4952 testing images on 20 categories of indoor and outdoor objects class. A total of 1434 images

containing cars in training set and testing set in the PASCAL VOC 2007 dataset are extracted to be evaluated. The car detection evaluation criterion is the same as PASCAL object detection. Intersection over Union (IoU) is set as 0.7 to assess a correct localization.

4.2. Implementation Details. All experiments are conducted based on mm detection [36], which is an open-source object detection toolbox based on PyTorch. Following [4, 7], this paper resized each image so that its shorter side has 600 pixels. The network is trained using one Nvidia RTX 3070 GPU with 1 image per minibatch. This paper uses a weight decay of 0.0001 and a momentum of 0.9. The learning rate started from 0.001 and was divided by 10 at every five epochs. The model had 10 total epochs. All other hyperparameters in this paper follow mm detection [36].

4.3. Main Results

4.3.1. Experiments on the KITTI Test Set. For evaluating the performance of the proposed framework, this paper first trains the network with all the KITTI training data and then tests it on the KITTI testing set. The results are compared with recent results for vehicle detection on the KITTI testing set. Table 1 presents the detection results of all methods on the three categories of the KITTI dataset. As shown in Table 1, the proposed method achieves the best average precision in both categories: "easy" and "moderate." To be more specific, the proposed framework surpasses the Faster R–CNN with FPN backbone by 4.73%, 8.04%, and 8.12% in "easy," "moderate," and "hard" test settings, respectively. The proposed method also surpasses other object detection frameworks, including one-stage frameworks such as YOLOv2 and MS-CNN and two-stage frameworks such as SINet [25] and improving Faster R–CNN [26]. Comparing with multitask CNN [20], the proposed network achieves better results in "easy" and "moderate" test settings by 2.07% and 0.51%, respectively. However, multitask CNN achieves the best AP in "hard" test setting. "Hard" group contains vehicles with bounding boxes sizes smaller than 25 pixels and invisible vehicles that are difficult to see with the naked eye. Multitask CNN introduced an efficient voting mechanism to refine the score of each RoI and the subcategory-aware nonmaximum suppression to tackle the occlusions better; thus this framework is more efficient in locating vehicles that are small and occluded. In the future, this paper will look at some methods to handle small and occluded vehicles better. For the inference speed, the proposed model takes 0.13 seconds for processing an image. Compared with SINet [25], the proposed model achieves comparable inference speed, while outperforming in terms of the AP in all groups.

Figure 6 presents some visualizations of detection results of the proposed method (shown in the right column) and the faster R-CNN with FPN backbone (shown in the left column) on the KITTI testing set. As shown in this figure, the proposed network can locate exactly vehicles in difficult conditions. It is also clear that more accurate boxes are generated by the proposed network and more small and occluded vehicles have been detected.

4.3.2. Experiments on the PASCAL VOC 2007 Car Dataset. This paper also compares the proposed network on the PASCAL VOC 2007 car dataset with several competitive models, including fast R-CNN [38], faster R-CNN [1], DPM [39], and multitask CNN [20]. This paper mainly focuses on the AP of car class appearing in images. There are a total of 1434 images containing cars in this dataset, and the ratio between training and testing images is 1 : 1. Table 2 provides a comparison of the detection performance of these five methods on car class of the PASCAL VOC 2007 dataset. The AP obtained by the proposed method is 78.84%, which outperforms all other methods by a large margin. The experimental results demonstrate that the proposed method can accurately locate vehicles, though the dataset is very small.

Furthermore, this paper also uses floating point operations (FLOPs) to calculate the computational cost of the proposed model on the PASCAL VOC 2007. Recent general object detectors are adopted to evaluate the computational cost of each network, including YOLOv2 [3], YOLOv3 [40], and faster R-CNN [4]. The experimental results of each network are shown in Table 3. All results are reported based on mm detection [36]. The GFLOPs of the proposed model is 73.2 which is comparable with the results of YOLOv3 and faster R-CNN. However, the proposed approach achieves significant improvements on vehicle detection compared to YOLOv2 and faster R-CNN as shown in Tables 1 and 2. The results demonstrate that the proposed model is simple and efficient. In Section 4.3.3, this paper conducts ablation experiments to show the advantages of each proposed module in reducing the computational cost and improving the detection performance.

4.3.3. Ablation Experiments on the KITTI Validation Set. This paper performs ablation analysis of the proposed method on the KITTI validation set to evaluate how different components affect the detection performance. Table 4 shows the experimental results. First, this paper replaces the original RPN in the faster R-CNN with FPN backbone with the proposed improving RPN. The improving RPN is applied to all feature maps generated by the FPN backbone {$P2$, $P3$, $P4$, $P5$} for generating proposals with different scales and aspect ratios. As shown in Table 4, the improving RPN module achieves comparable results compared with the results of the original RPN while providing better runtime speed. In this paper, the improving RPN is designed to generate feature map containing more discriminative information to facilitate the enhanced feature maps generated by the multilayer enhancement module. Next, the multilayer enhancement module is adopted to reweights feature representations of feature maps generated by the FPN backbone to increase the discrimination of foreground objects and background regions in each feature map. The multilayer enhancement module takes all feature maps generated by the FPN backbone {$P2$, $P3$, $P4$, $P5$} and intermediate feature

Table 1: Detection results of the proposed method and other methods on the KITTI testing set. The inference time is evaluated on single Nvidia RTX 3070 GPU per image.

Method	Average precision (%)			Inference speed (s)
	Easy	Moderate	Hard	
Faster R-CNN [1]	86.71	81.84	71.12	0.84
YOLOv2 [3]	76.79	61.31	50.25	0.01
Faster R-CNN with FPN backbone [4]	88.62	84.14	73.20	0.92
MS-CNN [37]	90.03	89.02	76.11	0.18
Improving faster R-CNN [26]	89.20	87.86	74.72	0.06
SINet [25]	89.60	90.60	77.75	0.12
Multitask CNN [20]	91.28	91.67	**85.43**	—
Proposed method	**93.35**	**92.18**	81.32	0.13

(a) (b)

Figure 6: Detection results of the proposed method (a) and the Faster R–CNN with FPN backbone (b) on the KITTI testing set.

Table 2: Detection results of the proposed method and other methods on the PASCAL VOC 2007 car dataset.

Method	Average precision (%)
Fast R-CNN [38]	52.95
Faster R-CNN [1]	59.82
DPM [39]	57.14
Multitask CNN [20]	63.91
Proposed method	**78.84**

maps produced by the improving RPN as inputs. The detection results are shown in Table 4. The multilayer enhancement module dramatically improves the detection performance. To be more specific, the multilayer enhancement module improves the AP by 3.5%, 3.37%, and 2.1% in "easy," "moderate," and "hard" group, respectively. Finally, an adaptive RoI pooling module is added to generate RoI

Table 3: Comparative results of GFLOPs with different models.

Model	Input resolution	GFLOPs
YOLOv2 [3]	416 × 416	17.4
YOLOv3 [40]	416 × 416	62.8
Faster R-CNN [4]	~600 × 1000	57.9
Proposed model	~600 × 1000	73.2

Table 4: The AP results and the inference time of each enhanced module and the original Faster R–CNN with FPN backbone.

Model	Average precision (%)			Time/image (s)
	Easy	Moderate	Hard	
Faster R–CNN with FPN backbone	89.52	87.14	78.20	0.17
+Improving RPN	89.21	88.27	78.12	0.09
+Improving RPN + multilayer enhancement module	92.71	91.64	80.22	0.11
+Improving RPN + multilayer enhancement module + adaptive RoI pooling	**93.67**	**92.08**	**82.41**	0.13

The inference time is evaluated on single Nvidia RTX 3070 GPU.

features from different pyramid levels. The adaptive RoI pooling module includes CARoI pooling to pool features from different levels and max operation to fuse different features. The results are shown in the last row of Table 4. As shown, the adaptive RoI pooling module further enhances the detection performance while keeping the efficiency of the network. Particularly, the improvements on "Hard" group, which contains many small vehicles, are more significant. This result indicates that the adaptive RoI pooling module is more efficient in detecting small vehicles.

5. Conclusion

This paper develops a novel framework based on multiscale feature learning and enhanced feature pyramid for vehicle detection. For generating RoIs and intermediate feature maps, an improving region proposal network is introduced. A novel multilayer enhancement module is designed to generate more discriminative representation feature maps. By improving multiscale features with strong semantics, the performance of vehicle detection has been substantially improved. In addition, an adaptive RoI pooling module is designed to better exploit RoI features from different pyramid levels and produce a better RoI feature for the detection network. Experiments that evaluate the performance of the proposed method are carried out on both the KITTI testing set and the PASCAL VOC 2007 car dataset. Compared with the state-of-the-art methods on vehicle detection, the proposed method demonstrated better experimental results. Furthermore, experiments that evaluate the vehicle detection system are carried out on the KITTI validation set. The experiments results demonstrated the effectiveness of different components in the proposed framework. In the future, this paper will look at some methods to handle small and occluded vehicles better.

Data Availability

The codes used in this paper are available from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Amsterdam, The Netherlands, October 2016.

[3] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger,," in *Proceedings of the CVPR*, pp. 1–9, Honolulu, HI, USA, July 2017.

[4] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Honolulu, HI, USA, July 2017.

[5] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, Italy, October 2017.

[6] P. Xiong, K. He, E. Q. Wu, L. T. E. Zhu, A. Song, and P. X. Liu, "Human exploratory procedures based hybrid measurement fusion for material recognition," *IEEE/ASME Transactions on Mechatronics*, 2021.

[7] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "Augfpn: improving multi-scale feature learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12595–12604, Seattle, WA, USA, June 2020.

[8] H. Nguyen, "Fast traffic sign detection approach based on lightweight network and multilayer proposal network," *Journal of Sensors*, vol. 2020, Article ID 8844348, 13 pages, 2020.

[9] B. Chen, M. Xia, and J. Huang, "Mfanet: a multi-level feature aggregation network for semantic segmentation of land cover," *Remote Sensing*, vol. 13, no. 4, p. 731, 2021.

[10] M. Xia, Y. Cui, Y. Zhang, Y. Xu, J. Liu, and Y. Xu, "DAU-Net: a novel water areas segmentation structure for remote sensing image," *International Journal of Remote Sensing*, vol. 42, no. 7, pp. 2594–2621, 2021.

[11] M. Xia, T. Wang, Y. Zhang, J. Liu, and Y. Xu, "Cloud/shadow segmentation based on global attention feature fusion residual network for remote sensing imagery," *International Journal of Remote Sensing*, vol. 42, no. 6, pp. 2022–2045, 2021.

[12] N. C. Mithun, N. U. Rashid, and S. M. M. Rahman, "Detection and classification of vehicles from video using multiple time-spatial images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1215–1225, 2012.

[13] A. Ottlik and H.-H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner-city intersections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 211–225, 2008.

[14] S. Kyo, T. Koga, K. Sakurai, and S. Okazaki, "A robust vehicle detecting and tracking system for wet weather conditions using the IMAP-VISION image processing board," in

Proceedings of the ITSC, pp. 423–428, Tokyo, Japan, October 1999.

[15] J. Cui, F. Liu, Z. Li, and Z. Jia, "Vehicle localisation using a single camera," *In Proc. 2018 IEEE Intelligent Vehicles Symposium*, pp. 871–876, Jun. 2010.

[16] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*vol. 33, no. 3, pp. 514–530, 2011.

[17] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition," in *Proceedings of the IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 6–20, February 2014.

[18] R. M. Z. Sun and G. Bebis, "Monocular precrash vehicle detection: features and classifiers," in *Proceedings of the IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2019–2034, September 2006.

[19] W. C. Chih-Wei Cho and C. W. Cho, "Online boosting for vehicle detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 892–902, 2010.

[20] W. Chu, Y. Liu, C. Shen, D. Cai, and X.-S. Hua, "Multi-task vehicle detection with region-of-interest voting," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 432–441, 2018.

[21] Y. Cai, Z. Liu, X. Sun, L. Chen, H. Wang, and Y. Zhang, "Vehicle detection based on deep dual-vehicle deformable Part Models," *Journal of Sensors*, vol. 2017, Article ID 5627281, 10 pages, 2017.

[22] X. Li, Y. Liu, Z. Zhao, Y. Zhang, and L. He, "A deep learning approach of vehicle multitarget detection from traffic video," *Journal of Advanced Transportation*, vol. 2018, Article ID 7075814, 11 pages, 2018.

[23] H. Wang, X. Lou, Y. Cai, Y. Li, and L. Chen, "Real-time vehicle detection algorithm based on vision and lidar point cloud fusion," *Journal of Sensors*, vol. 2019, Article ID 8473980, 9 pages, 2019.

[24] X. Yuan, S. Su, and H. Chen, "A graph-based vehicle proposal location and detection algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3282–3289, 2017.

[25] X. Hu, X. Xu, Y. Xiao et al., "SINet: a scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, 2018.

[26] H. Nguyen, "Improving Faster R-CNN framework for fast vehicle detection," *Mathematical Problems in Engineering*, vol. 2019, Article ID 3808064, 11 pages, 2019.

[27] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," *Clinical Orthopaedics and Related Research*, https://arxiv.org/abs/1704.04861, 2017.

[28] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9190–9200, Long Beach, CA, USA, June 2019.

[29] S. Liu, Q. Lu, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.

[30] M. Tan, R. Pang, and V. Quoc, "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790, Seattle, WA, USA, June 2020.

[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Honolulu, HI, USA, July 2017.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proceedings of the CVPR*, pp. 3354–3361, Providence, RI, USA, June 2012.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[35] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 924–933, IEEE, Santa Rosa, CA, USA, March 2017.

[36] K. Chen, J. Pang, J. Wang et al., *Chen Change Loy, and Dahua Lin. Mmdetection*, https://github.com/open-mmlab/mmdetection, 2018.

[37] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the Computer Vision - ECCV 2016*, pp. 354–370, Amsterdam, The Netherlands, October 2016.

[38] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[40] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.