Vehicle Object Detection Based on Improved RetinaNet

Luyang Zhang¹, Haitao Wang¹*, Xinyao Wang¹, Shuai Chen¹, Huai-bin Wang¹, Kai Zheng²,
Hailong Wang³

¹College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China.
²Special Equipment Safety Supervision Inspection Institute of Jiangsu Province, Nanjing 210036, China
³Jiangsu Mingyuan Rail Transit Equipment Co. LTD. Nanjing, 210044, China.

*875727548@qq.com

Abstract. Aiming at the low efficiency of vehicle object detection in real scenes, this paper proposes an improved RetinaNet. An octave convolution structure and a weight pyramid structure are introduced respectively to improve the detection performance of RetinaNet for vehicles. Specifically, we use octave convolution instead of the traditional convolution layer to improve the feature map's representation of detailed information. In addition, in order to improve the quality of feature fusion, a weighted feature pyramid network (WFPN) structure is proposed to limit the propagation of gradients between different levels. The experimental results on the DETRAC data set show that the method has good detection results for vehicle targets of different scales in different scenarios, and can better meet the needs of practical applications.

Keywords: Vehicle detection, RetinaNet, Octave convolution, WFPN

1. Introduction

With the increasing number of cars, traffic supervision has become a challenging problem. Vehicle object detection is a key link in Intelligent Transportation System (ITS) and plays an important role in the normal operation of the entire system. Therefore, it has always been a research hotspot for scholars at home and abroad. However, the detection accuracy is low in real scenes due to the complex environment, low resolution, etc.[1].

In recent years, with the promotion of deep learning technology, computer vision technology has been greatly improved. The research of target detection technology has also entered an unprecedented stage of development. Object detection algorithms based on deep learning are usually divided into two categories: Among them, one is to detect targets through classification area suggestions, such as RCNN[2], Fast-RCNN[3], Faster-RCNN[4], etc., This type of algorithm is called a two-stage method. First, a sliding window method is used to obtain a candidate area on the picture, then the feature vector of the candidate area is extracted, and finally, the scoring result of the classifier is used to determine the target category of the candidate area. The other is one-stage methods such as YOLO[5], SSD[6], etc. which use a method of directly returning to the target category. Although the detection speed is
greatly improved, the accuracy is sacrificed. After a period of time, people are exploring the balance between detection accuracy and speed. As a one-stage detector, RetinaNet [7] introduces a new focal loss function and a feature pyramid network (FPN) to make it reach the accuracy of the most advanced two-stage method while ensuring the detection speed. Become one of the most popular detection algorithms.

However, RetinaNet is limited by its backbone network and still has poor performance in dealing with complex environments and low-resolution problems. Therefore, this paper proposes an improved RetinaNet. In order to improve the performance of the backbone network, we replace the traditional CNN with an octave convolution structure [8] to improve the representation of the detailed information of the picture; Then in order to solve the problem of gradient migration in the process of multi-scale feature extraction, we propose a weighted feature pyramid machine structure. Inspired by the gated structure [9], the low-level features are used as the base layer, and the adjacent high-level features are used as the auxiliary layer. The feature score module adaptively adds weights to the auxiliary layer, which improves the offset problem in the gradient propagation process. The experimental results on DETRAC dataset show the effectiveness of the method.

2. Methodology

In this section, we will introduce our method in detail, including the overall network architecture, octave convolution, and weighted feature pyramid network.

2.1. RetinaNet

RetinaNet is mainly composed of three parts: the backbone network network (ResNet) for feature extraction, feature pyramid networks (FPN), classification and regression sub-networks. The image features are initially extracted by ResNet, reorganized in FPN, and finally sent to the classification and regression sub-network to obtain the final detection results.

2.2. OctaveConv module

The image information contains both high frequency part and low frequency part. The high frequency part mainly contains detailed information, while the low frequency part usually reflects the general information of the image. However, the traditional convolution structure is to perform convolution operation by shifting the sliding window, and the distribution of high frequency and low frequency is ignored. In order to solve this problem, proposed OctaveConv to replace the convolutional layer on traditional CNN. The octave convolution module divides the feature map into a high-frequency channel and a low-frequency channel, and then the feature map of the low-frequency channel is scaled down to half of the original. Finally, the feature map is divided into high frequency part and low frequency part.
The input feature map $X$ can be divided into high frequency $X^H$ and low frequency $X^L$. Perform convolution operations from high frequency to high frequency and from high frequency to low frequency on $X^H$ respectively, and the obtained feature maps are respectively denoted as $Y^{H-H}$ and $Y^{H-L}$. Among them, $Y^{H-L}$ is generated by the high frequency part after averaging pooling down-sampling to half the original size, and then convolution operation; $Y^{L-H}$ is the result of upsampling the low frequency part to double the size of the feature map, and then convolution. The final high-frequency and low-frequency feature maps are as follows:

$$
Y^H = Y^{H-H} + Y^{L-H}
$$
$$
Y^L = Y^{L-L} + Y^{H-L}
$$

(1)

Octave convolution structure of our design include the initial layer, transition layer and output layer:

**Initial layer**: The function of the initial layer is to extract the high-frequency feature map and low-frequency feature map of the input image ($X^H$ and $X^L$ in Fig 2). The input image is passed through a convolutional layer with a convolution kernel size of 3x3 to generate high-frequency feature maps. Low-frequency feature maps are obtained in the same way, but add an average pooling layer before the convolutional layer. The number of channels of the high-frequency and low-frequency feature maps depends on the parameter $\alpha$. If the number of input channels is expressed as $\text{channels}$, the number of low-frequency channels is $\text{channels} \times \alpha$, and the number of high-frequency channels is $\text{channels} \times (1-\alpha)$. The value of $\alpha$ ranges from 0 to 0.5, and the value of $\alpha$ in this paper is 0.25.

**Transition layer**: The transition layer is to reprocess the feature map. $X^H$ and $X^L$ first go through a convolutional layer, and then perform up-sampling and down-sampling operations respectively. Finally, new $Y^H$ and $Y^L$ are generated by summing.

**Output layer**: The role of the output layer is opposite to that of the initial layer. The output $Y^L$ of the transition layer is upsampled to the same size as $Y^H$ after passing through a convolution layer, and then added to the feature map obtained after $Y^H$ passes through a convolution layer. Finally, the output feature map of the octave convolution module is obtained.

**2.3. Weighted feature pyramid network (WFPN)**

ResNet is used as the backbone network of RetinaNet and uses FPN to extract multi-scale feature maps. The formation of the FPN feature map first needs to perform an up-sampling operation on high-level features, and then sum with adjacent low-level features. However, the summation operation may cause unexpected gradient propagation. The detailed description is shown in Fig 3, P2 is the sum of

![Diagram](image-url)
the low-level feature C2 and the up-sampled high-level feature C3, then the derivation of P2 to the unknown variable may be the reciprocal of C2, or the derivative of C3 to the weight of C2 or C3. What’s more, when performing a summation operation between two feature layers, the loss of the accumulated features of the lower level will be directly transferred to the features of the higher level during the back propagation of the gradient.

A typical situation is shown in Fig 3(a). The gradient propagation from P2 to C3 has completely dominated, resulting in the training process not converging in the right direction. The desired result is that the gradient propagates from P2 to C2 to occupy an absolutely dominant position (Fig 3(b)). Therefore, we propose a weighted pyramid network structure that enables the network to adaptively determine the main contribution points of the backbone feature map, avoiding spatial shifts and accidental propagation of gradients.

![Fig. 3 Schematic diagram of gradient propagation migration](image)

Specifically, the first two main feature maps are divided into a base layer and an auxiliary layer; Then the two feature layers are mapped to a specific feature score through a shared Valid convolution layer, and the weight factors of the base layer and the auxiliary layer are determined according to the feature scores. Finally, the sigmoid function is used to normalize the weight factor to the range of 0 to 1, and the weight score and the feature layer are multiplied and summed to form the feature map in FPN.

![Fig. 4 Weight pyramid structure](image)

Fig 4 shows the weight pyramid structure we designed. C_i and C_{i+1} represents the base layer and auxiliary layer respectively; C_i^* represents the feature map of the base layer after the 1×1 convolutional layer, and C_{i+1}^* is the feature map of the auxiliary layer after the 1×1 convolutional layer and up-sampling. m_i and m_{i+1} represent the feature scores of the base layer and auxiliary layer mapping respectively. We retain all the information of the base layer (weight is 1), and add weight ω to the auxiliary layer, which can be expressed as:

\[
ω_i = \frac{m_i - m_{i+1}}{m_i}
\]

\[
P_i = C_i^* + ω_i C_{i+1}^*
\]
\[ P_i \] is the final output feature map. Since feature scores are learned adaptively, we believe that they play an important role in preventing the accidental propagation of spatial displacement and gradient.

3. Experiment
This article uses Pytorch as the backend and implements the proposed model in mmdetection. We verify the performance of the method in the DETRAC data set (960×540), which contains 84 K for the training set and 56 K for the test set. In order to verify the training process in real time, we divide the training set into two parts, where 56K is used for training and 28K is used for verification.

In order to ensure the fairness of the experiment, all parameter settings follow the settings in the baseline method. Our experiment was performed on a GTX 1080Ti graphics card (12G memory) with a batch size of 4, and the model was optimized using stochastic gradient descent (SGD). In particular, 12 epochs are set in the training process, and the initial learning rate is 0.01, which decreases to 50% of the current learning rate as the epoch increases.

This paper designs three experiments to verify the effectiveness of the proposed method, adding octave convolution, WFPN, and both to RetinaNet respectively. The experimental results are shown in Table 1. It can be seen that the model after adding octave convolution can greatly improve the detection of difficult data; The addition of the WFPN module can improve the model's detection performance in low-resolution scenes such as night and rain. Not surprisingly, applying both methods to RetinaNet has achieved good results, which not only improves the detection accuracy of difficult data, but also shows certain advantages in dealing with low-resolution problems.

| Method       | Overall | Easy  | Medium | Hard  | Sunny | Cloudy | Rainy | Night |
|--------------|---------|-------|--------|-------|-------|--------|-------|-------|
| RetinaNet[8] | 68.87   | 89.65 | 73.12  | 53.64 | 83.73 | 72.42  | 53.40 | 73.93 |
| +OctaveConv  | 69.87   | 90.12 | 74.16  | 58.04 | 83.60 | 72.84  | 54.29 | 74.22 |
| +WFPN       | 69.14   | 89.87 | 72.96  | 54.53 | 84.25 | 74.89  | 55.97 | 75.42 |
| +Both       | 71.62   | 90.42 | 75.59  | 57.74 | 84.08 | 74.86  | 56.21 | 75.09 |

Table 1 Performance evaluation on detrac dataset

4. Conclusion
In order to improve RetinaNet's detection performance of vehicle targets, we designed an octave convolution structure and WFPN structure. Our method is to replace the traditional CNN with octave convolution in the feature extraction module to improve the model's ability to characterize high-frequency detail features. In addition, a weighted feature pyramid machine structure is proposed to solve the gradient migration problem in the process of multi-scale feature extraction. The feature score module adaptively adds weights to the auxiliary layer of the backbone feature mapping, which effectively improves the offset problem in the gradient propagation process. The experimental results on the DETRAC dataset confirm that our method can not only improve the detection accuracy of difficult data, but also shows certain advantages in dealing with low-resolution problems.

Acknowledgment
This work was supported by the Science and Technology Project of Jiangsu Provincial Market Supervision Bureau, China (KJ185609)

References
[1] Wei L, Shengcai L and Weidong H 2019 Towards accurate tiny vehicle detection in complex scenes J. Neurocomputing 347 pp24-33
[2] Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation. Int. Conf. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[3] Girshick R 2015 Fast R-CNN *Int. Conf. The IEEE International Conference on Computer Vision* (ICCV)

[4] Shaoqing R, Kaiming H and Girshick R 2015 Faster R-CNN towards real-time object detection with region proposal networks *J. Advances in Neural Information Processing Systems* pp 91-99

[5] Redmon J and Divvala S 2016 You only look once unified real-time object detection *Int. Conf. IEEE Conference on Computer Vision and Pattern Recognition* pp779-788

[6] Liu W 2016 SSD Single shot MultiBox detector *Int. Conf. Eur Conf Comput Vis* pp21-37

[7] Lin T Y, Goyal P, Girshick R, He K and Dollar P 2018 Focal loss for dense object detection. *IEEE J. Trans. Pattern Anal. Mach. Intell*

[8] Yunpeng C, Haoqi F and Bing X 2019 Drop an octave Reducing spatial redundancy in convolutional neural networks with octave convolution *IEEE International Conference on Computer Vision* pp3435-3444

[9] Jin Y, Hengchang X, and Yi X 2019 Gated CNN Integrating Multiscale Feature Layers for Object Detection *J. Pattern Recognition* 105 pp107-131