International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 04 – Issue 07 || July 2019 || PP. 92-98

Vehicle Detection Based on 3D Point Cloud under the Framework of Improved Faster R-CNN

***Mengchao Liu[1], Zhuangzhuang Mao[2], Hongpeng Zhou[3]**

School of Mechanical and Power Engineering
Henan Polytechnic University, 454003, Jiaozuo, China
** Corresponding author*

Abstract: At present, the Faster R-CNN is widely used in the field of vehicle detection. Because vehicle detection based on the RGB image is greatly affected by the environment. So in this paper, a vehicle detection method based on the 3D point cloud is proposed. We transformed the 3D point cloud data acquired by Velodyne 64 lines 3D lidar into the depth map, then linear interpolation and smooth denoising are carried out on the depth maps, and a new depth map dataset is obtained. Then, after the anchor value of RPN (Region Proposal Networks) is modified, the improved Faster R-CNN+VGG16 can be used to complete the training and testing of the dataset under the framework of Caffe. The experimental result shows that the improved Faster R-CNN+VGG-16 achieves average precision (AP) 63.89% in vehicle detection based on the 3D point cloud.

Keywords: Improved Faster R-CNN; Vehicle detection; RPN; 3D point cloud; depth map

1. Introduction

The vehicle detection based on the Convolutional Neural Networks (CNN) has become more and more common. From the R-CNN to the Faster R-CNN, the application of CNN in vehicle detection is becoming more and more extensive. In 2016, He Kai Ming *et al*. proposed the Faster R-CNN. Considering the structure of the Fast R-CNN, they designed the Region Proposal Networks (RPN) specifically for extracting Region Proposals [1]. It completes a leap from 2s to 10ms in the time of feature extraction of each image, greatly improves the detection accuracy and speed. VGG-16 basically inherited the AlexNet structure. The AlexNet uses only eight layers, while the two versions of the VGG have 16 layers and 19 layers respectively. VGG-19 and VGG-16 have almost the same accuracy, but the operation of VGG-16 is Faster [2]. In this paper, the VGG-16 model combined with the Faster R-CNN are used for completing the vehicle detection tasks. In order to improve the effect of vehicle detection based on the 3D point cloud, we combined the Faster RCNN and VGG16 to complete the vehicle detection work. Compared with the vehicle detection based on the RGB image, the vehicle detection based on the 3D point cloud has its own advantages, for example, fast processing speed, low memory consumption, and image acquisition is not affected by light and weather [3]. Therefore, in order to avoid the disadvantages of the vehicle detection based on the RGB image, we proposed a vehicle detection based on the 3D point cloud. In order to improve the vehicle detection accuracy, the anchor of the Faster R-CNN is modified.

2. Network Structure

The network structure of VGG-16 is composed of 13 Convolution Layers, 5 MaxPooling Layers and 3 Fully-Connected Layers, the structure of it can be shown in Figure 1. Conv is the Convolution Layer, and FC is the Fully-Connected Layer. The convolution kernel of 3×3 with step size of 1 is used in the Convolution Layer

to extract the Feature Maps [4]. After the input image passes through the Convolution Layer, the Feature Maps are obtained. If the Feature Maps are classified directly, the computation will be too heavy to carry out normal experiments. At this point, the Max Pooling Layer can reduce the features and parameters in the calculation process. The categories of targets in the input image are classified by the Fully-Connected Layer [5].

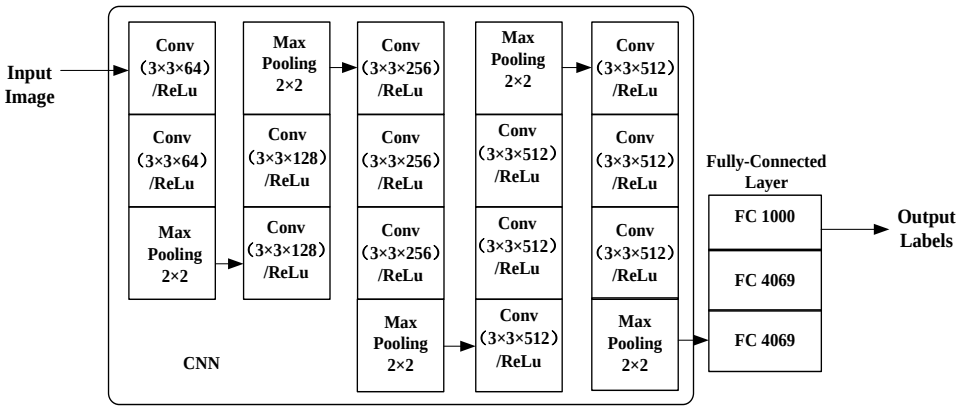

Figure 1. The structure of VGG-16

The Faster R-CNN can be divided into four parts [6]. The first part is the CNN, which is used to extract the Feature Maps. It includes 13 Conv Layers, 13 ReLu Layers and 4 Pooling Layers. The second part called RPN is used to extract Region Proposals [7]. The third part is the ROI Pooling Layer, which integrates the Region Proposals extracted by RPN and the Feature Maps produced by CNN to generate the Proposal Feature Maps. The Classification Layer, the fourth part, sends the Proposal Feature Maps into the Fully-Connected Layer, and then the category of each proposal is calculated by the SoftMax Classification Function, at the same time Bounding Box Regression was calculated by the RPN Loss Function. More accurate bounding boxes can be obtained. The structure of the Faster R-CNN is shown in Figure 2, which can be regarded as the combination of the Fast R-CNN and RPN [8].

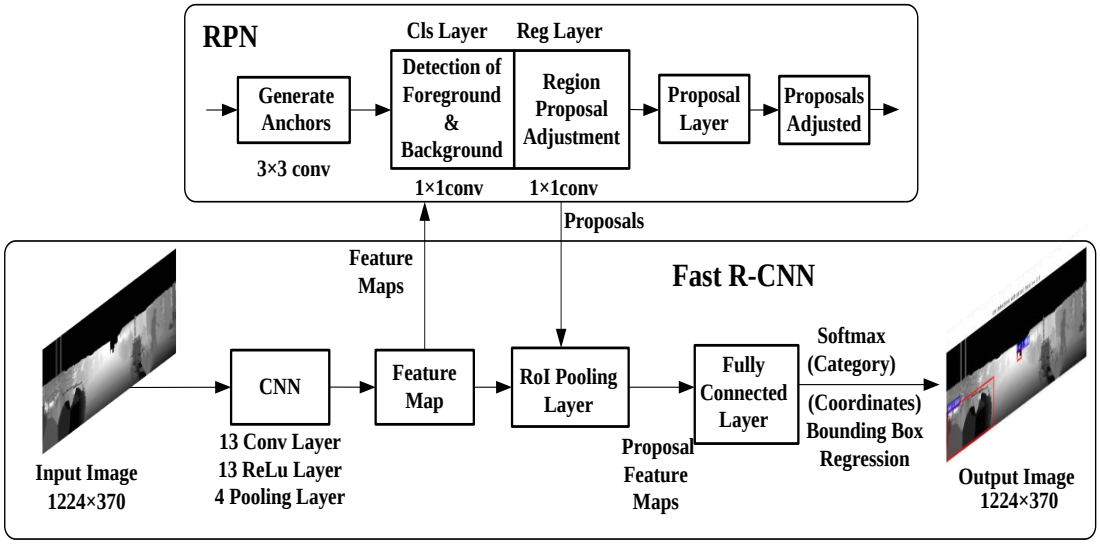

Figure.2 The structure of Faster R-CNN

RPN is used to extract the Region Proposals [9]. RPN is equivalent to generating many candidate anchors in the scale of the input image. Then CNN is used to judge which ones are foreground anchors with target and background anchors without target. The background anchors are abandoned. The Bounding Box

Regression for foreground anchors is carried out by the RPN Loss Function, and more accurate proposals can be obtained. The RPN Loss Function is a sum of the Classification Loss and the Bounding Box Regression Loss.

Anchor is an important concept of RPN. RPN generates a lot of anchors on the Feature Maps. After 3×3 convolution on the Feature Maps, each pixel point is mapped back to the corresponding coordinate point of the input image [10]. With this point as the center, three kinds of candidate areas of different sizes, namely "anchor", are generated with three proportions of (1,0.5,2) and three scales of (128/256/512) respectively. The anchor generation mechanism is shown in Figure 3.

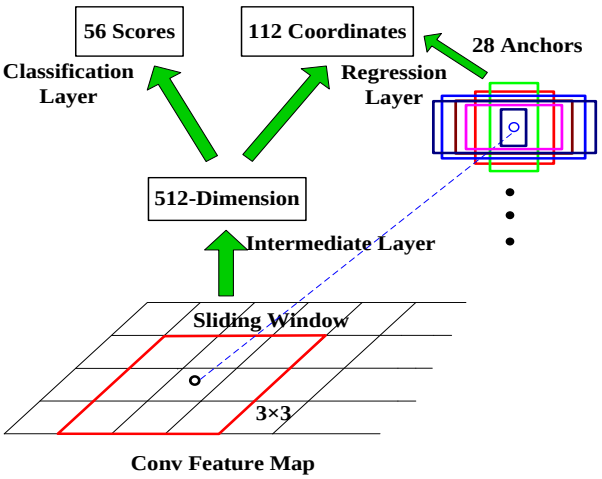

Figure.3 Anchor generation mechanism

There are 12,857 rectangle boxes for all the images in the dataset. Figure 4 is the histogram of rectangle boxes amount distribution with different aspect ratios and areas. In order to improve the detection accuracy, we change the aspect ratio and scales of anchor to (0.2,0.35,0.5,1,2,3,4) and (64,128,256,512) respectively.

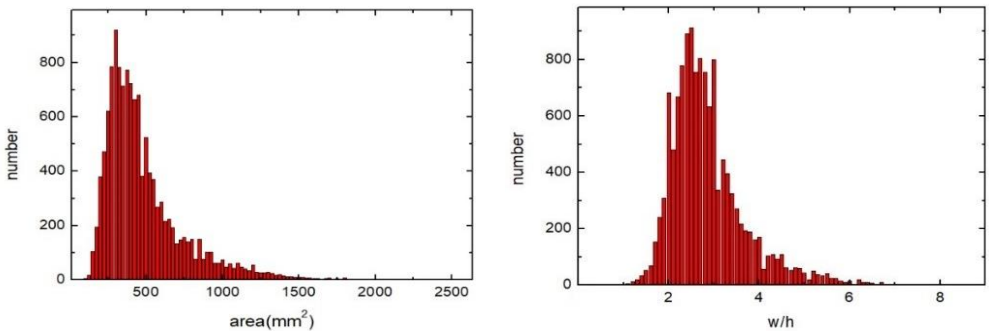

Figure.4 Histogram of rectangle box amount distribution with different aspect ratios and area

3. Experimental Dataset

We process 1500 3D point cloud information files collected by Velodyne 64 lines 3D lidar in KITTI dataset to completed the transformation of the 3D point cloud into depth map. The point cloud data (PCD) is

projected and filtered into the image field. When dealing with the nearest pixel value, pixels without a depth value are replaced by pixels from adjacent. These pixels are then sparsely mapped. Then sparse mapping is carried out on these pixel points to obtain depth map. Then the interpolation and the smoothing denoising for the depth maps is carried out. Finally the normalized depth maps are obtained, and we save them in a folder to generate 1500 depth images with the size of 1224×370. Then, in the Ubuntu system, the LabelImage is used to label objects with rectangle boxes. After then, the 1500 xml format files are generated, which correspond to 1500 depth map in png format.

The training of the Faster R-CNN is based on the model already trained. In practice, the training process can be divided into six steps [11]:

Step1: RPN is trained for the first time on the already trained model. (stage1_rpn_train.pt)

Step2: The region proposals are collected using RPN trained in the first step. (rpn_test.pt)

Step3: The first time to we train the Fast R-CNN. (stage1_fast_rcnn_train.pt)

Step4: The second time we train the RPN. (stage2_rpn_train.pt)

Step5: RPN trained in step 4 is used to extract region proposals. (rpn_test.pt)

Step6: The second time we train the Fast R-CNN. (stage2_fast_rcnn_train.pt)

The processing and training of the dataset is show in Figure 5.

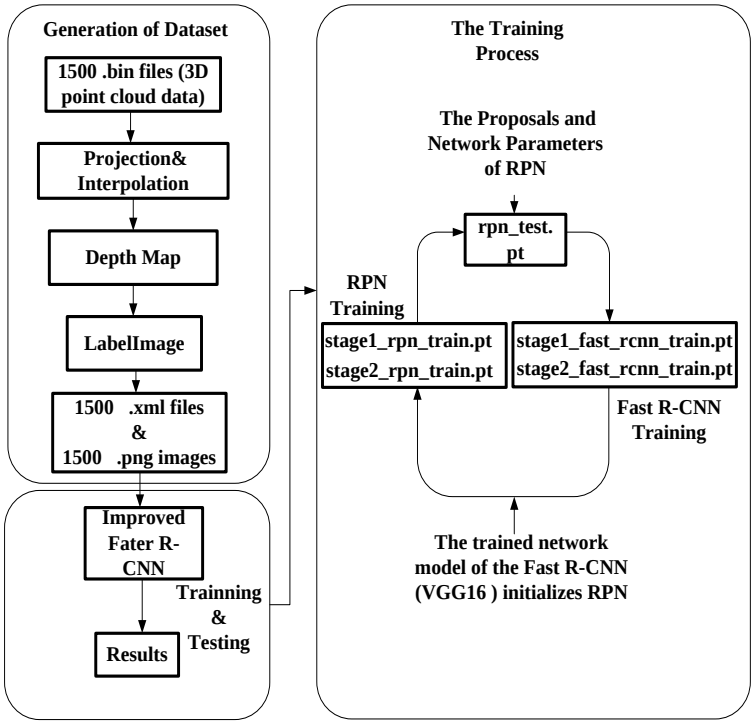

Figure.5 The processing and training of the dataset

4. Analysis of Experimental Results

In the experiment, the CNN is implemented in Caffe framework. The experimental hardware configuration is Intel Core i7 processor with 8G NVIDIA GeForce GTX 1070 GPU for training. The operating system is Ubuntu18.04 platform and the programming environment is based on Python. 1200 pictures of the dataset are used for training and 300 pictures for testing. The output probability value of the Classification Layer

uses the threshold value of 0.7 to perform the Non-Maximum Suppression. The maximum number of iterations of RPN and the Faster RCNN is set to 80,000 and 40,000 times respectively. Experimental result shows that the combination of the Faster R-CNN + VGG-16 model can achieve better effect on AP of the vehicle detection. There is no great difference in detection time before and after network improvement. Table 1 shows the AP comparison of the Faster R-CNN before and after improvement. The runtime is the average time it takes to detect an image.

Table.1 AP value of Faster R-CNN +VGG16 model

Model	*Object*	*AP*	*Runtime(s)*
Faster RCNN+VGG16	Car	59.10%	0.059
Improved Faster RCNN +VGG16	Car	63.89%	0.063

According to scene A and scene B in Figure 6, we can get good detection effect whether it is a small target or a large target. The detection results of these two scenes are based on the improved Faster R-CNN, while the detection result of scene C in Figure 7 is based on the Faster R-CNN before improvement. According to the test result, we can know that the detection effect of the networks after the improvement on distance vehicles is enhanced. The average detection time of each image during detection is show in Table 1.

Figure.6 Scene A and B

Figure.7 Scene C

The following is the P-R curve of the experimental evaluation index, where the precision can be expressed as:

$$precision = \frac{TP}{TP+FP} \qquad (1)$$

Recall rate can be expressed as：

$$Recall = \frac{TP}{TP+FN} \qquad (2)$$

where positive cases are correctly decomposed into positive cases, represented by TP. Positive cases are correctly classified into negative cases, represented by FN. Negative cases are correctly classified into negative cases, represented by TN. Negative cases are correctly classified into positive cases, represented by FP [12]. AP is the area under the PR curve, which can be expressed as formula (3). As shown in Figure 8, the Recall of the improved Faster R-CNN+VGG16 model is enhanced. This proves that the accuracy of vehicle detection has increased.

$$AP = \int_0^1 P(R)\,dR \qquad (3)$$

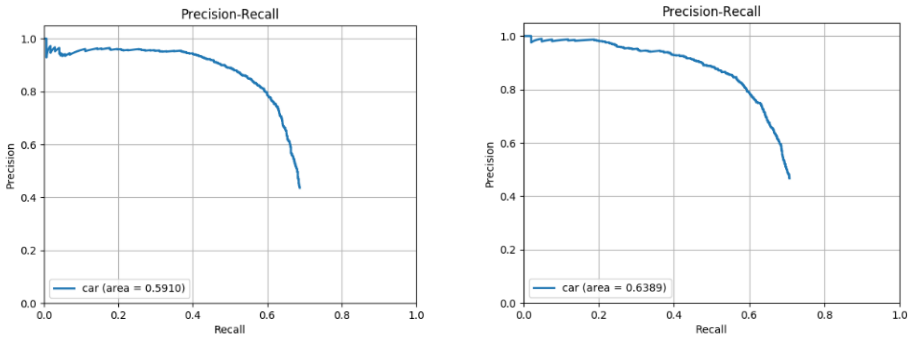

Figure.8 Comparison of the P-R curve before and after network improvement (Faster R-CNN+VGG16)

5. Conclusion

The experimental results show that the vehicle detection based on the 3D point cloud has a good effect. In terms of object detection speed, the CNN can process depth maps faster. Vehicle detection based on the 3D point cloud is unaffected by light and weather in image acquisition. For example, in dark, vehicles in front of the detection vehicle can still be detected, which cannot be achieved by vehicle detection based on the RGB images. However, vehicle detection based on the 3D point cloud needs to transform the point cloud data into the Depth

Maps, and the production of the dataset is slightly more troublesome than that of RGB dataset. In the future, vehicle detection based on the 3D point cloud can be used as an auxiliary detection method in extreme light or harsh conditions.

Reference

[1] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region prop osal networks[C]//Advances in neural information processing systems. pp. 91-99,2015.

[2] Zhou Y, Tuzel O.Voxelnet: End-to-end learning for point cloud based 3d object detection [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp.4490-4499, 2018.

[3] Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network[J]. arXiv preprint arXiv:1608.07916, 2016.

[4] Chen X, Gupta A. An implementation of Faster R-CNN with study for region sampling[J]. arXiv preprint arXiv:1702.02138, 2017.u555

[5] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv.org*. November 11.

[6] Li J,Liang X, Shen S M, et al. Scale-aware fast R-CNN for pedestrian detection[J]. IEEE transactions on Multimedia, pp985-996,2017

[7] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[8] Sun X, Wu P, Hoi S C H. Face detection using deep learning: An improved faster RCNN approach[J]. Neurocomputing, pp.299: 42-50,2018

[9] Ren Y, Zhu C, Xiao S. Object detection based on Fast/Faster RCNN employing fully convolutional architectures[J]. Mathematical Problems in Engineering, 2018.

[10] Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network[J]. arXiv preprint arXiv:1608.07916, 2016.

[11] Roblek D, Szegedy C, Jurewicz J S. Object detection using neural network systems: U.S. Patent Application 15/650,790[P]. 2019-1-17.

[12] Muhovič J, Bovcon B, Kristan M, et al. Obstacle Tracking for Unmanned Surface Vessels Using 3-D Point Cloud[J]. IEEE Journal of Oceanic Engineering, 2019.