World Scientific Research Journal Volume 6 Issue 3, 2020
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

Improved Faster RCNN Object Detection

Chaojie Li[1, a], Peng Zhou[1]

[1]College of Information Engineering, Shanghai Maritime University, China.

[a]hans.lcj@foxmail.com

Abstract

Aiming at the problem that the low accuracy of Faster RCNN object detection algorithm, an improved stereo object detection method based Faster RCNN (Stereo Faster RCNN, S-Faster RCNN) is proposed and used to the vehicle detection. A two-layer feature extraction network is used to extract the left and right image features respectively, and then the features are connected and fed into the mixed RPN network to correlate the slight differences between the left and right images and train the RPN. The features of the associated left and right images will be conducted RoI Pooling operation respectively to form feature maps of fixed size. Finally, the corresponding categories of objects and the accurate location of the bounding boxes will be output through the full connected layers. The experimental results show that compared with the traditional Faster RCNN algorithm. The accuracy of the improved method improves by 11.3%, 9.5% and 6.3% respectively under the three standards of KITTI dataset: simple, medium and difficult.

Keywords

Object detection; Faster RCNN; RPN; vehicle detection.

1. INTRODUCTION

Object detection is an important computer vision research task, which is mainly used to deal with the classification and localization of certain types of visual objects (such as people, animals or cars) in digital images [1,2,3,4]. Target detection algorithms mainly include two categories:

one-stage methods and two-stage methods. One-stage methods are also called regression-based method, mainly including YOLO, SSD, etc. [5-6]; the two-stage methods are also called candidate box-based method, mainly including R-CNN, Fast R-CNN, Faster RCNN, etc. [7-9]. The One-stage methods are dominant in speed, while two-stage methods are have higher accuracy. However, most of the above methods are improved from the perspective of data and parts of the original network, they do not take full advantages of the stereo vision accurate positioning characteristics and the stereo multi-information inputs, which similar to human eyes, so the detection results are relatively limited.

Based on the idea of stereo inputs, this paper expands Faster RCNN by adding an extra viewpoint of input to simultaneously detect and associate object in left and right images. The proposed method makes full use of the small differences of the images in the left and right perspectives to train the RPN network. Compared with monocular image input, stereo image inputs enrich object information and improve detection accuracy. The experimental results on the vehicle detection KITTI dataset show that the recall and accuracy of the method in this paper are greatly improved compared with the original Faster RCNN algorithm.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

2. S-FASTE RCNN

2.1. The Structure of Proposed Method S-Faster RCNN

S-Faster RCNN adds another input to the monocular Faster RCNN to expand the monocular image input to stereo image inputs. Compared with the original network, the new model contains four parts: stereo feature extraction, RPN, RoI (Region of Interests) Pooling, and prediction. The improved network is shown in Figure 1.

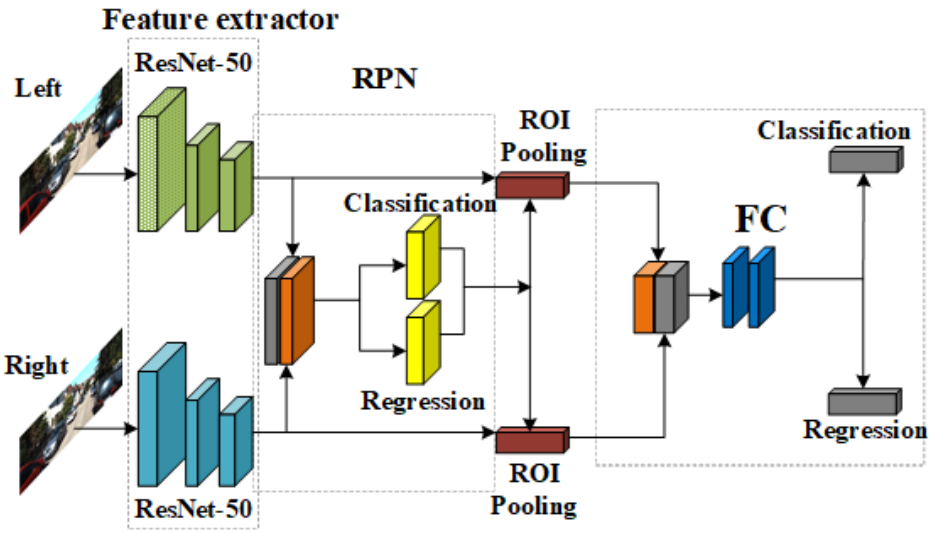

Figure 1. S-Faster RCNN

The detection process is as follows: The stereo RGB images are used as two inputs to enter the feature extraction network to obtain stereo image feature maps. Then the feature maps will be connected and fed into the RPN to train RPN classification and regression networks. After the RoI pooling operation, we will get series of candidates of bounding box, then feed these RoIs into FC (full connected layer) to classify and regress the bounding boxes, getting the target categories and the precise position of the final bounding box.

2.2. RPN

The original RPN takes the monocular feature map as input and cannot handle the case of stereo inputs. This article makes the following modifications for stereo input: Connect the features of the left and right stereo feature maps before RPN network training.

2.2.1. RPN Classification

Since the same object appears twice in the left and right perspectives of stereo images, the left and right images alone cannot represent the characteristics of the entire object. Therefore, the object truth box (Ground-Truth Box, GT Box) needs to be redefined as follows: The classification truth box is the union of left and right image truth boxes, as shown in Figure 2.

The union ground- truth box is represented as follows:

$$\mathrm{GT}_{Union} = GT_{left} \cup GT_{right} \tag{1}$$

Where $\mathrm{GT}_{left}, GT_{right}$ respectively represent left and right ground-truth boxes.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

Because of the connection of left and right image features, when calculating the intersection-over-union (IoU) of the bounding box with the ground truth box, we need to modify the definition of IoU, the new IoU calculation formula is as follows:

$$\mathrm{IoU} = \frac{Box_{left\_right} \cap GT_{Union}}{Box_{left\_right} \cup GT_{Union}} \tag{2}$$

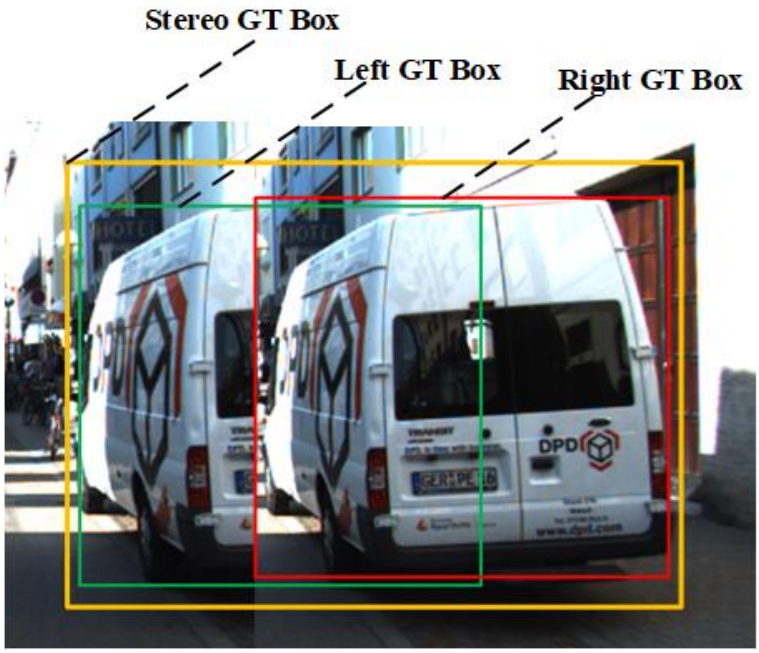

Figure 2. Classification GT Box

Where Box_{left_right} indicates the predicted bounding box after feature map fusion.

The predicted positive and negative labels are defined as follows

$$\begin{cases} Positive, IoU > 0.7 \\ \mathrm{Negtive}, IoU < 0.3 \end{cases} \tag{3}$$

The positive sample of the prediction is set to 1, otherwise it is set to 0. The loss function is a logarithmic loss function and is defined as follows:

$$\mathrm{L}_{cls}(p_i, p_i^*) = -\log(p_i p_i^* + (1-p_i^*)(1-p_i)) \tag{4}$$

Where p_i is the predicted label and p_i^* is the ground- true value.

2.2.2. RPN regression

In the case of the original monocular image, the training RPN bounding box is a 4-parameter form: (x, y, w, h),while we need to return to 8 parameters. However, in the actual situation, the left and right images' heights are the same, and the input stereo images have been aligned and calibrated. It can be considered that the vertical coordinates of the center points of the left and

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

right images are consistent with the width of the images' vertical direction. Based on this, the regression parameters are reduced from 8 to 6:

$$\mathrm{P}=\{x_{left}, w_{left}, x_{right}, w_{right}, y, h\} \tag{5}$$

The regression loss function is $\mathrm{smooth}_{\mathrm{L}_1}$ loss function, which needs to be adjusted to a 6-parameter form. The new regression formula is as follows:

$$L_{reg}(t_i, t_i^*)=\sum_{i\in P} smooth_{L_1}(t_i, t_i^*) \tag{6}$$

The loss function formula is:

$$smooth_{L_1}=\begin{cases} 0.5x^2, |x|<1 \\ |x|-0.5, |x|\geq 1 \end{cases} \tag{7}$$

The original Faster RCNN uses a 4-coordinate form when performing bounding box parameter regression:

$$\begin{cases} t_x=(x-x_a)/w_a, t_y=(y-y_a)/h_a \\ t_w=\log(w/w_a), t_h=\log(h/h_a) \end{cases} \tag{8}$$

$$\begin{cases} t_x^*=(x^*-x_a)/w_a, t^*{}_y=(y^*-y_a)/h_a \\ t^*{}_w=\log(w^*/w_a), t^*{}_h=\log(h^*/h_a) \end{cases} \tag{9}$$

New regression of prediction box:

$$\begin{cases} t_{x_{left}}=(x_{left}-x_a)/w_a, t_{x_{rightt}}=(x_{right}-x_a)/w_a \\ t_{y_{left}}=t_{y_{right}}=(y-y_a)/h_a \\ t_{w_{left}}=\log(w_{left}/w_a), t_{w_{right}}=\log(w_{right}/w_a) \\ t_{h_{left}}=t_{h_{right}}=\log(h/w_a) \end{cases} \tag{10}$$

$$\begin{cases} t^*{}_{x_{left}}=(x^*{}_{left}-x_a)/w_a, t^*{}_{x_{rightt}}=(x^*{}_{right}-x_a)/w_a \\ t^*{}_{y_{left}}=t^*{}_{y_{right}}=(y^*-y_a)/h_a \\ t^*{}_{w_{left}}=\log(w^*{}_{left}/w_a), t^*{}_{w_{right}}=\log(w^*{}_{right}/w_a) \\ t^*{}_{h_{left}}=t^*{}_{h_{right}}=\log(h^*/w_a) \end{cases} \tag{11}$$

$(x_{left}, y), (x_{right}, y)$ represent the center coordinates of the left and right bounding boxes respectively. $(w_{left}, h), (w_{right}, h)$ represent the length and height of the left and right images.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

The total RPN network loss function contains classification loss and regression loss:

$$L(\{p_i, p_i^*\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum p_i^* L_{reg}(t_i, t_i^*) \quad (12)$$

N_{cls}, N_{reg} are the normalization parameters, which are set to 256 and 2400 in our experiment. The parameter λ is to make the loss of classification and regression to the same proportion of total loss, set to 10.

Then, non-maximum suppression (NMS) is performed on the regions of interest of the left and right image features, reducing similar boxes, and selecting a certain number of boxes closest to the ground-true box. In this paper, we get the first 1000 highest scores for training, and the first 200 are taken for testing.

2.3. Classification and Regression

When sampling the RoIs, we consider a left-right RoI pair as foreground if the maximum IoU between the left RoI with left GT boxes is higher than 0.5, meanwhile the IoU between right RoI with the corresponding right GT box is also higher than 0.5. A left-right RoI pair is considered as background if the maximum IoU for either the left RoI or the right RoI lies in the [0.1, 0.5) interval. For foreground RoI pairs, we assign regression targets by calculating offsets between the left RoI with the left GT box, and offsets between the right RoI with the corresponding right GT box.

3. EXPERIMENTAL PROCESS

3.1. Experimental Platform

This experiment is completed in Linux environment. The operation system is Ubuntu 16.04 64 bits, the computer has 32.0GB of memory and Intel (R) Core (TM) i7-9700K CPU@3.60Ghz processor. The graphics card uses NVIDIA GeForceRTX2080Ti. All experiments are conducted based on Pytorch framework, with language of python.

3.2. Data Set

We evaluate our method on the challenging KITTI [10] object Detection benchmark. The dataset was collected from real images including cities, villages, and roads, with each image containing up to 15 vehicles. In this paper, all the 7481 training set images are roughly divided into training set and validation set in equal proportions.

3.3. Network Training

We use 5 different scale anchor boxes {32,64,128,126,512} with 3 ratios {0.5,1,2} to train the RPN network. In order to make the image size in the original dataset suitable for the network model proposed in this paper, the shortest side of the original image is first converted into a 600 pixel size. The classification regression input channel of the RPN network in Faster RCNN is 512. while stereo image input is used in this paper, so after the feature extraction, the input channel is now 1024. We train the network using SGD with a weight decay of 0.0005 and a momentum of 0.9. The learning rate is initially set to 0.001 and reduced by 0.1 for every 5 epochs. We train 20 epochs with 2 days in total.

4. RESULT ANALYSIS

In the experiment, we use recall and precision to evaluate the performance:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

$$recall = TP / (TP + FN)$$
$$precision = TP / (TP + FP) \quad (13)$$

The P-R(Precision-Recall) curves of Faster RCNN and our method under three different vehicle detection standards are shown in Figure 3 and Figure 4, respectively. From the comparison of Precision-Recall curves, it can be seen that compared with the original Faster RCNN, the S-Faster RCNN algorithm proposed in this paper has better accuracy and recall performance under the three standards of simple, medium and difficult.

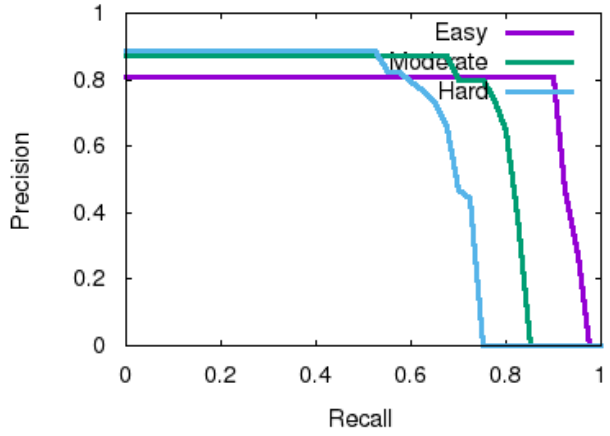

Figure 3. Faster RCNN

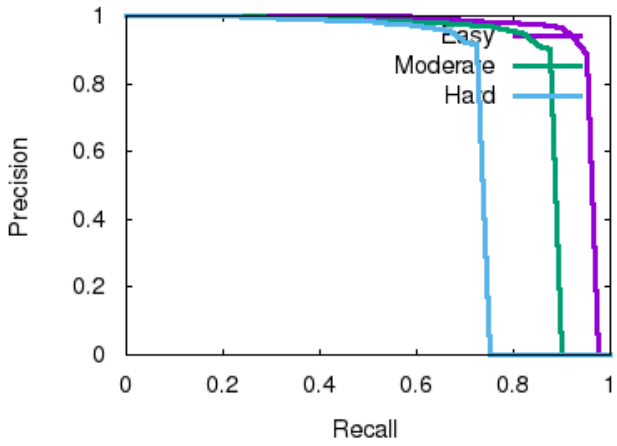

Figure 4. Method in This paper

Average Precision (AP) represents the area between the curve and the coordinate axis. According to KITTI vehicle detection requirements, when the overlap of the prediction box and the ground-truth box is higher than 0.7, the prediction is considered correct, otherwise the prediction is considered incorrect. In order to compare the impact of the basic feature extraction network on the final detection results, the experiment adds Faster RCNN detection results using VGG-16 as a feature extraction network. The average accuracy and running time are shown in Table 1.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

Table 1. AP and runtime of different methods

methods	AP(IoU=0.7)	Runtime/s
VGG16	75.3	0.22
ResNet50	78.9	0.24
Ours	90.2	0.29

Experiment shows that compared with Faster RCNN based on ResNet-50 feature extraction network, our method improves AP by 11.3% under the easy standard. The result of replacing only the feature extraction network shows that the final detection result of ResNet-50 as the feature extraction network is better than VGG-16. This article is based on stereo inputs, so the processing time is longer than Faster RCNN based on ResNet-50 (the processing time of a single image increased from 0.24 s to 0.29 s). Some visualization results are shown in Figure 5.

Figure 5. Visualization of some detection results

5. CONCLUSION

This paper proposes a stereo S-Faster RCNN algorithm based on the monocular Faster RCNN, which turns the original single image input into left and right stereo image inputs for feature extraction and trains a hybrid RPN network. The combined feature proposal regions are used for object classification and bounding box regression. Experiments on the vehicle detection data set KITTI show that the proposed S-Faster RCNN method has a significant improvement in recall, accuracy, and average accuracy compared to the original Faster RCNN algorithm, which proves the effectiveness of the proposed method. However, the improved method increases the complexity of the network and the detection time is increased compared to the original method. The next work will be the optimization of network structure and parameters to achieve better real-time detection.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202003_6(3).0010

REFERENCES

[1] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik,"Simultaneous detection and segmentation," in European Conference on Computer Vision. Springer, 2014, pp. 297–312.

[2] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.

[3] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel,"Image captioning and visual question answering based on attributes and external knowledge," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 6,pp. 1367–1381, 2018.

[4] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang,Z. Wang, R. Wang, X. Wang et al., "T-cnn: Tubelets with convolutional neural networks for object detection from videos," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2896–2907, 2018.

[5] JOSEPH R, SANTOSH D, ROSS G, et al. You only look once: Real-time object detection [C]//IEEE Conference on Computer Vision and Pattern Recongnition, Lasvegas, IEEE, 2016:27 - 30.

[6] LIU W, ANGUELOV D, ERHAN D, et al. Single shot multibox detector [C] // European Conference on Computer Vision, Amsterdam, Springer, 2016:21 – 37.

[7] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] //IEEE Conference on Computer Vision and Pattern Rec-ognition, Honolulu, IEEE, 2014:580 – 587.

[8] GIRSHICK R. Fast R-CNN [C] //EEE International Conference on Computer Vision, Venice, IEEE, 2017:3039 – 3048.

[9] REN S Q, HE K M, GIRSHICK R, et al. Faster RCNN: Towards real-time object detection with region proposal networks [C] // International Conference on Neural Information Processing Systems, Montreal, Neural Information Processing Systems Foundation, 2015:91 – 99.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. //In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3354–3361. IEEE, 2012.