Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 7, July 2019, pp. 1792-1801
DOI: 10.23940/ijpe.19.07.p5.17921801

Pedestrian Detection based on Faster R-CNN

Shuang Liu[a,*], Xing Cui[a], Jiayi Li[a], Hui Yang[a], and Niko Lukač[b]

[a]*School of Computer Science and Engineering, Dalian Minzu University, Dalian, 116605, China*
[b]*Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, SI-2000, Slovenia*

Abstract

Pedestrian detection has a wide range of applications, such as intelligent assisted driving, intelligent monitoring, pedestrian analysis, and intelligent robotics. Therefore, it has been the focus of research on target detection applications. In this paper, the Faster R-CNN target detection model is combined with the convolutional neural networks VGG16 and ResNet101 respectively, and the deep convolutional neural network is used to extract the image features. By adjusting the structure and parameters of Faster R-CNN's RPN, the multi-scale problem existing in the pedestrian detection process is solved to some extent. The experimental results compare the detection ability of the two schemes on the INRIA pedestrian dataset. The resulting model is migrated and validated on the Pascal Voc2007 dataset.

Keywords: pedestrian detection; faster R-CNN; feature extraction; deep learning

(Submitted on December 10, 2018; Revised on January 12, 2019; Accepted on February 8, 2019)

1. Introduction

Pedestrian detection is an essential branch of computer vision. It is widely used in artificial intelligence, driver assistance systems, intelligent robots, intelligent video surveillance, human behavior analysis, intelligent transportation, and other fields. However, pedestrians have the characteristics of rigidity and flexibility, and the appearance is susceptible to wearing scale, occlusion, posture, or viewing angle. Pedestrian detection is still a hot topic in the field of computer vision that is both challenging and research-worthy.

Pedestrian detection methods mainly include background modeling and approaches based on statistical learning. Background modeling mainly extracts foreground moving targets and performs feature extraction in the target area. Then, the extracted features are classified by a classifier, and the classification result is used to determine whether there is a pedestrian. However, there are still some problems in the background modeling method, such as environmental changes, cameras shaking when shooting pictures, and incorrect detection in dense scenes. The statistical learning method is the current mainstream method and constructs a pedestrian detection classifier by using samples. The classifier mainly includes Support Vector Machines (SVM), AdaBoost, and neural networks. Since the performance of the classifier is significantly affected by the training samples, the negative samples cannot cover all real application scenarios during offline training.

Dalal et al. proposed a pedestrian detection method based on Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM) in CVPR2005 [1]. In [2], Walk proposed an improved HOG algorithm using the HIK SVM classifier, namely the HOF and CSS (Color Self Similarity) features. In [3], Dollar used the idea that features of similar images can be accurately estimated. In 2010, BMVC proposed a theory that only needs to train a standard model. In [4], the DPM algorithm was applied to target detection, and the DPM model was used to detect pedestrians with severe adhesions, achieving good results [5]. With the development of deep learning, a series of excellent object detection frameworks have emerged, such as R-CNN (Region-CNN (Convolutional Neural Network)) [6], SPPNet (Spatial Pyramid Pooling Convolutional Networks) [7], Fast R-CNN [8], and Faster R-CNN [9].

There have been many advances in pedestrian detection using deep learning methods, but there are still certain problems to be solved in a complex scene. For example, it is difficult to separate pedestrians from backgrounds mixing or

* Corresponding author.
E-mail address: 19499080@qq.com

different persons are occluding with each other. In the environment of public stations, squares, and streets, pedestrians are moving, staying still, changing postures, or in different mutual occlusion status. These problems have caused difficulties for detecting pedestrians precisely. An improved method is put forward in this paper to complete the feature extraction network and the Region Proposal Network (RPN) [9] by using the Faster R-CNN object detection framework. Pedestrian detection performance is improved for cases of occlusion or small targets. Finally, the improved method is transferred to the Pascal Voc2007 dataset to test its generalization ability in a multi-target environment.

2. Related Work

2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a multi-layer neural network. Through a series of methods, it continuously reduces the image recognition problem of large data volume and finally enables it to be trained.

A typical convolutional neural network consists of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 1. The convolution layers cooperate with the pooling layers to form a plurality of convolution groups. Features are extracted layer by layer and are finally classified by some fully connected layers. It can argue that the concept of local receptive fields inspires the operations performed by the convolutional layer. Pooling layers are mainly applied in dimension reduction. In conclusion, CNN is distinguished by convolution simulation features. The convolution weight sharing and pooling reduce the order of the network parameters. Finally, classification and other tasks are performed through traditional neural networks.

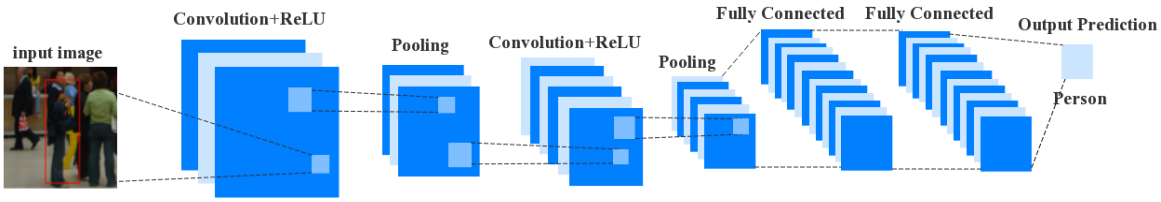


Figure 1. The basic structure of CNN

2.2. Original Faster R-CNN

After the accumulation of RCNN and Fast RCNN, Girshick proposed Faster R-CNN in 2016. Structurally, Faster R-CNN integrates extracted features, extracted regional proposal network proposals, frame regression, and classification into a network. The comprehensive performance has been dramatically improved, especially regarding detection speed. The basic frame structure of Faster R-CNN is shown in Figure 2.

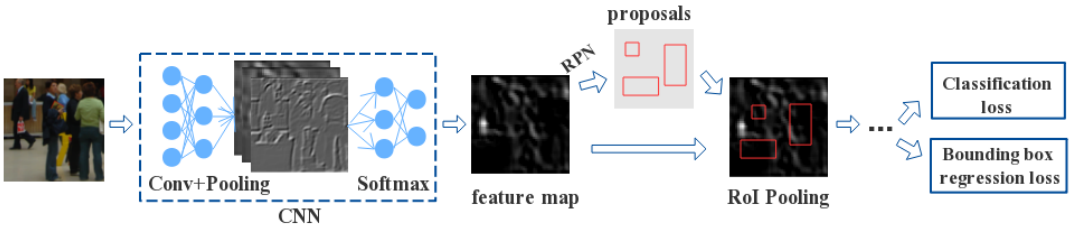


Figure 2. Faster R-CNN structure framework

As a general framework for object detection, Faster R-CNN mainly consists of two parts. They are responsible for extracting the RPN module of Region Proposal and the Fast R-CNN detection module. The function for the image is defined as Equation (1).

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

Where N_{cls}, N_{reg} represent the normalized parameters of classification and regression, respectively. L_{cls}, L_{reg} indicate the

loss of classification and regression, respectively. λ is the balance weight. p_i, p_i^* are the predicted value and the true value of the target candidate box, respectively. t_i, t_i^* are the predicted value and true value of the detected category, respectively. Compared to traditional methods, Faster R-CNN differs in that it replaces regional proposals with deep convolutional neural networks. Region Proposal Network (RPN) is more efficient and is a core part of the algorithm. RPN changes the way that traditional sliding window methods generate Region Proposal. The Proposal and CNN classifications are combined to achieve a complete end-to-end deep convolutional neural network target detection model.

As shown in Figure 2, an image is input, and the feature map and RoI are extracted from it by using the "Convolutional Neural Network + Pooling Layer" structure. The approximate location of the target is obtained using the network training method after extracting the feature map. Then, it continues to train the approximate location of the acquisition to make them more accurate. After obtaining the precise position, the softmax classifier and candidate box regression are used to obtain the category and target frame regression of the detected object. The method puts Fast-RCNN and RPN in the same network structure to train and share network parameters. RPN can quickly extract high-quality Proposal, which not only speeds up target detection but also improves the detection performance.

3. Our Detection Method

3.1. Detection Algorithm

As a general target detection framework, Faster R-CNN has achieved good detection results and has real-time performance. For some pedestrian detection cases, the general framework is not entirely suitable. Therefore, the frame structure and parameters need to be adjusted for specific problems. Equations (2) to (5) are algorithm descriptions. An image x containing pedestrian is input into a deep neural network that extracts features. A feature map f_1 is generated and is sent to the RPN network to generate a pedestrian region proposal RP_x. The generated region proposals are obtained by the RoI mapping method, and then f_2 and f_3 are obtained. Finally, the selected pedestrian area is refined and sent to the fully connected layer f_c defined in Equation (6) for classification to realize classification and regression.

$$f_1 = Conv^{(1)}(x, w_1) \tag{2}$$

$$RP_x = RPN(f_1, w_{RP}) = [RP_x^{cls}, RP_x^{reg}] \tag{3}$$

$$f_2 = Conv^{(2)}(f_1, w_2) \tag{4}$$

$$f_3 = RoI^{(3)}(f_2, RP_x) \tag{5}$$

$$y = f_c[f_3, w_3, refine(RP_x)] = [RP_x, class] \tag{6}$$

There are five steps for pedestrian detection scheme designed in this paper, as shown in Figure 3.

Step 1 A test image is input.

Step 2 The entire image is input into a deep convolutional neural network for feature extraction.

Step 3 The improved RPN network is used to generate region proposals, which are then mapped to the last layer of convolution layers, that is, the feature map.

Step 4 A fixed size feature map is generated for each RoI by the RoI Pooling layer.

Step 5 The probability of classification and the candidate box regression are co-trained by the classification loss function Softmax Loss and the frame regression loss function Smooth L1 Loss.

3.2. Deep Convolution Network Structure Selection

The pedestrian detection method based on Faster R-CNN firstly adopts a series of "conv + relu + pooling" structure to

propose the feature maps of the input image. The feature maps will be used for follow-up RPN layers and the ROI Pooling process. The trained weights are the initial values for all layers. Based on the existing network weights, the last layer of the convolutional layer is fine-tuned and the network is retrained using the pedestrian dataset.

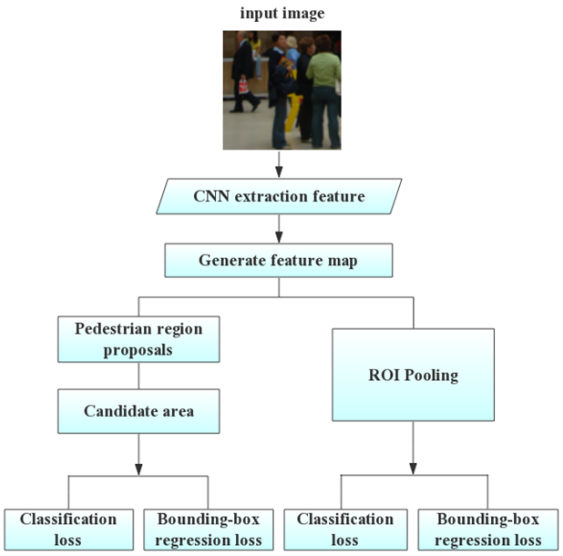


Figure 3. Pedestrian detection program

In this paper, different depth convolutional neural networks are used to extract image features. The VGG16 [10] used has a total of five convolutional layers, while each convolutional layer also contains two to three convolution kernels. A max pooling layer follows each convolution. After the convolutional layer, three fully connected layers are connected and a softmax classification layer is finally connect. The network structure and parameters are shown in Figure 4. Feature maps of the middle layer for VGG16 are shown in Figure 5.

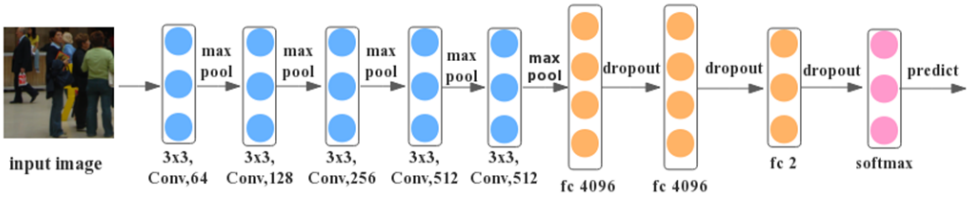


Figure 4. Schematic diagram of a VGG16 network structure

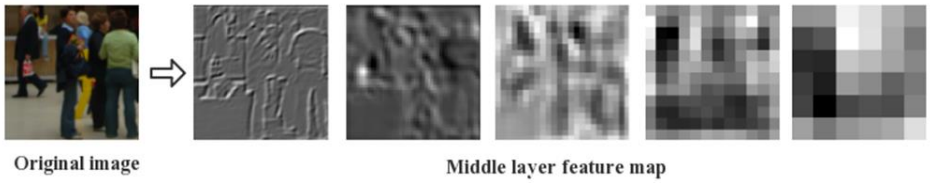


Figure 5. Visualization of the feature map for VGG16 middle layer

As the feature extraction network, the input pedestrian image is first normalized. The convolution kernel's size of the convolutional layer in the network is 3×3 with a step size of 1 and padding parameter of 1. In the hidden layer, a relu function for nonlinear optimization is set. The formula is $relu(x) = \max(0, x)$.When extracting image features, VGG16's full convolutional layer network is used. The output characteristics of the Conv5/Conv5_1 layer are obtained as the input of the softmax layer. For the detection task of this paper, the object of detection is the pedestrian in the image, so it can also be regarded as the distinction between pedestrian and background dichotomy.

In addition to using the deep convolutional VGG16 network, the experiment also utilized the ResNet [11] network. This type of network uses a connection called “shortcut connection” instead of a simple convolution stack. In this way, the problem of gradient disappearance caused by the deep network layer is effectively solved. ResNet network uses Conv1~Conv4_x for feature extraction. The final output of Conv4_x is the shared part of RPN and RoI Pooling. Network structure and parameters are shown in Figure 6.

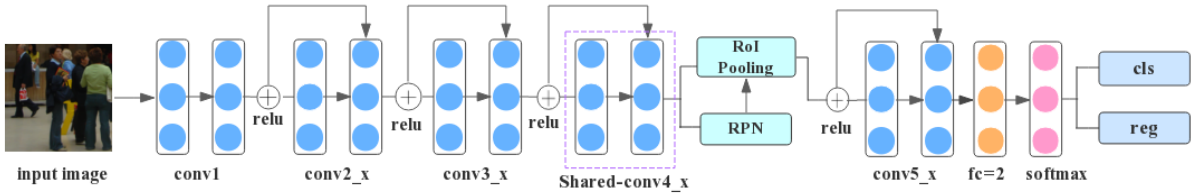


Figure 6. Faster R-CNN based on ResNet101

The feature map generated by RoI Pooling acts on Conv5_x and then connects to a fully connected layer to classify and return the feature map. Since the image mean file used by a particular detection scheme will be different from the data mean on ImageNet, the mean file of the pedestrian dataset will be recalculated. Then, it modifies the parameters of the last layer of the fully connected layer and modifies other settings.

3.3. Construction of Pedestrian Detection Network

The pedestrian detection network consists of two parts, wherein the RPN is a full convolutional neural network for extracting pedestrian region proposals. The RPN takes a feature map generated by the feature extraction network as an input and outputs a rectangular frame of the target region. Fast R-CNN uses the region proposals extracted by RPN to detect and identify the target. Although the traditional Faster R-CNN model has universal adaptability due to the position of pedestrians in the image, the size of the box is also different when generating candidate boxes. In order to improve the multi-scale detection capability of the pedestrian detection model, the RPN was improved in the experiment, as shown in Figure 7.

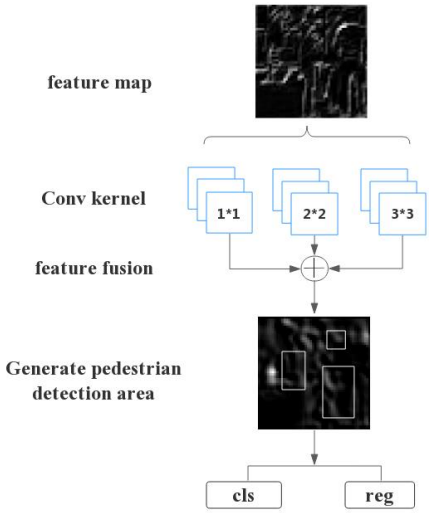


Figure 7. Improved RPN structure using different scale convolutions for sliding operations

For small targets in the image, the sizes of the candidate boxes are 64, 128, and 256. Pedestrians have different postures in various scenes, so the sizes displayed in the image are also different. To cover more of the detection targets in the image, the ratios are set to 1:2, 1:1, and 2:1. A total of nine different pedestrian candidate boxes are generated at each pixel. For a convolutional feature with an input width of W and a height of H, the total number of candidate boxes generated is N. The formula is shown as Equation (7).

$$N_{anchors} = W_{conv} \times H_{conv} \times K_{anchors} \tag{7}$$

In CNN, a point on the feature map corresponds to the size of the area on the input map, which is called the receptive field [12]. In order to increase the receptive field to improve performance, three methods are proposed in the literature called Convolutional Pose Machines [13]. The original RPN generates candidate regions only by convolution of one scale after extracting the features. This experiment is inspired by the related work and adds the sliding window operation to the last layer of the RPN feature map, using 1×1, 3×3, 5×5 convolution kernels. A better candidate region is obtained by using these different scale convolution operations to obtain receptive fields of different sizes.

The RPN phase finally retains the pedestrian region proposal candidate box. These candidate boxes use the position correspondence principle to the feature map of Conv5_3 output. Fixed size boxes are generated through the RoI Pooling operation. To make RPN and Fast R-CNN share the weight of the feature network, an alternate training method is employed. RPN extracts the proposal. The extraction result is input into the detection network Fast R-CNN. Fast R-CNN identifies the target detection in the pedestrian region proposals. Fast R-CNN extracts the feature maps using VGG16 and ResNet101's Conv1-Conv5 respectively. After the fifth layer convolving, the feature map size is 512×512. Then, it connects the fully connected layers of different dimensions, updates the network layer unique to the RPN, and retrains. The pedestrian detection network realizes the function of predicting the position of the candidate frame and detecting the target.

4. Experimental Results and Analysis

4.1. Training and Test Datasets

The training and test datasets in this paper come from the INRIA dataset. The INRIA dataset is currently the most used static pedestrian detection database. The training set has 614 positive samples (including 2,416 pedestrians) and 1,218 negative samples. The test set has 288 positive samples (including 1,126 pedestrians) and 453 negative samples [1]. The images in the file 90×160×H96/Train/pos are selected as the positive training samples. The images in the neg file under the Train directory are used as negative training samples. There are multiple pedestrian scenes in the dataset, including single and numerous people. Most of the human body is in a standing position, and the height is more than 100 pixels.

The image in the INRIA dataset has high resolution, but the background is more complicated. There are various shooting conditions, such as light changes, human body occlusion, etc. To prevent over-fitting due to the small number of pictures, the image augmentation technique is utilized in the experiment. Dataset expansion is obtained for the same target at different angles by performing operations such as rotation, translation, and scaling to a certain extent. In addition to using the pedestrian dataset, the experiment also selects Pascal Voc2007 as the migration dataset to verify whether the improved Faster R-CNN has good mobility.

4.2. Experimental Platform Settings

Experimental settings are listed in Table 1.

Table 1. Experimental platform environment configuration

Name	Configuration
Operating System	Ubuntu 16.04
CPU	Intel Xeon E5-1603v3 2.8 10M 1866 4C CPU
Memory	8G
GPU	Nvidia Quadro K1200
Parallel Computing Platform	CUDA 6.0,CUDNN8.0
Deep Learning Framework	Tensorflow
Programming Environment	Python2.7

4.3. Training Model

This paper uses a deep learning framework Tensorflow. The experiment selects 80% of the INRIA pedestrian dataset as the training set and 20% as the test set. The network for extracting features is separately used for pre-training VGG16 and ResNet101 on ImageNet. In the improved Faster R-CNN method proposed in this paper, the multi-scale convolution generation candidate region is performed on the RPN part, so the RPN layer of the trained Faster R-CNN model is changed into multi-scale convolution. The other parameters are unchanged. The classification layer and regression layer of the RPN are retrained. When training with the pre-training model, the ownership weight is randomized and then retrained according to the INRIA dataset. The model trained by the above steps can be directly used to detect pedestrian targets in the image.

Mean Average Precision (mAP) is used as an indicator to evaluate pedestrian performance in experiments, where Q is the number of categories detected and P is the accuracy. In the experiment, it is indicated how many recognized pedestrian areas are correct. R is the recall rate, which represents how many of the marked pedestrian areas have been detected. The area value formed by the accuracy rate and the recall rate is AP. mAP and AP are listed in Equations (8) and (9), respectively.

$$\mathrm{m}AP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{8}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{9}$$

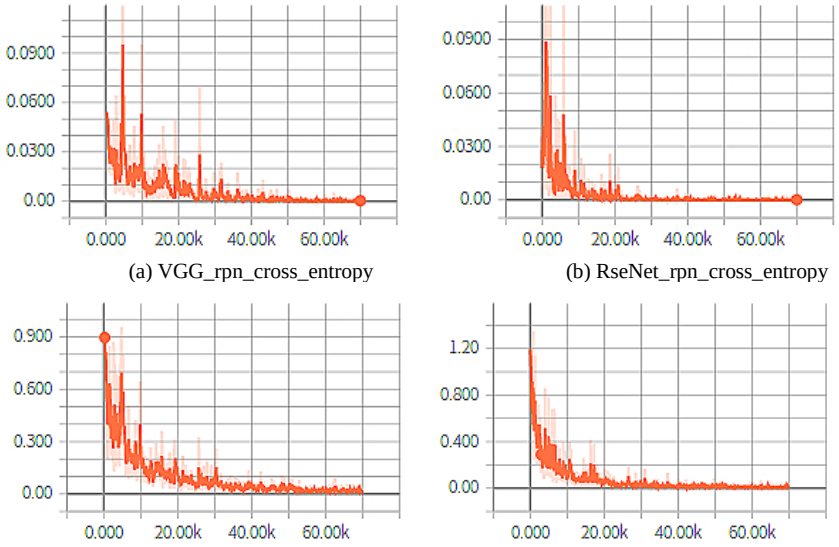
The training learning rate is set as 0.001 and the number of iterations as 70,000. When training Faster R-CNN, the number of images iterated each time into the training network is one. After scaling the image, the pixel size of the longest side of the image cannot exceed 1000, and the pixel size of the shortest side cannot be less than 600. The RoI overlap threshold is considered foreground, the foreground threshold is set to 0.5 during training, and the minibatch score marked as the foreground is 0.25. The pedestrian test results on the INRIA dataset are shown in Table 2.

Table 2. INRIA dataset to detect pedestrian test results

Method	Number of iterations	mAP (%)
Faster R-CNN+VGG16	70000	76.38
Faster R-CNN+ResNet101	70000	78.25

4.4. Analysis of Experimental Results

By using the improved Faster R-CNN model for training tests, mAP values of 76.38 and 78.25 are obtained under two different pre-training models for VGG and ResNet. Tensorboard is the Tensorflow's visualization tool that visualizes the running status of the Tensorflow program through log files output during the Tensorflow program running process. The cross entropy function curve and the loss curve of the training in Tensorboard2 are shown in Figure 8.



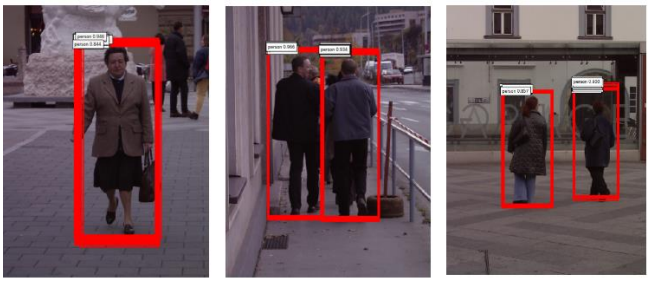
(a) VGG_rpn_cross_entropy (b) RseNet_rpn_cross_entropy

(c) VGG_total_loss (d) ResNet_total_loss

Figure 8. Cross entropy and loss curve for the experiment

It can be seen from Figure 8 that our model reached convergence after 70,000 iterations. This result is mainly due to the expansion of the experimental dataset and the expansion of the training samples. By improving the RPN part of Faster R-CNN, an anchor scale suitable for pedestrian detection is designed. A sliding window operation is added to the last layer of the RPN feature map. Convolution of different scales is used to obtain receptive fields of different sizes, and better candidate regions are obtained.

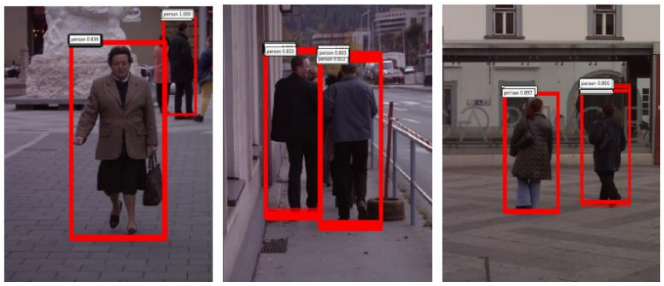
Figures 9 and 10 show the comparison of the improved results of Faster R-CNN combined with different networks. The test results on the test images are similar using these two network structures. However, for some smaller pedestrian targets and occlusion situations, it can be seen that using the ResNet101 network is better than using VGG16.



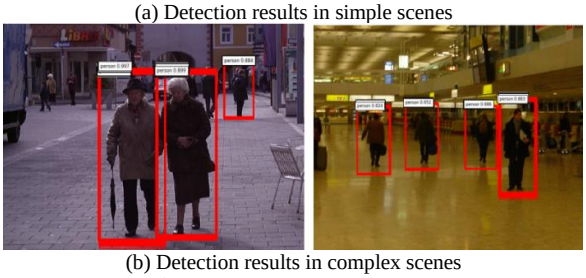
(a) Detection results in simple scenes

(b) Detection results in complex scenes

Figure 9. Pedestrian detection results of the improved Faster R-CNN combined with VGG16 network structure



(a) Detection results in simple scenes



(b) Detection results in complex scenes

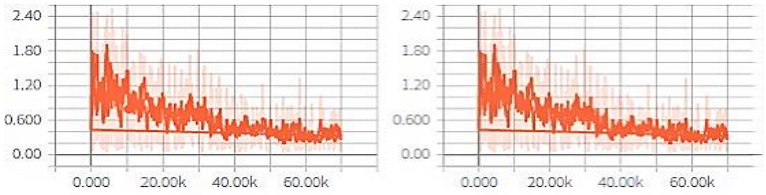
Figure 10. Pedestrian detection results of improved Faster R-CNN combined with ResNet101 network structure

Pascal Voc2007 is selected as the migration test set to test whether the improved Faster R-CNN has good generalization combined with different networks. The experiment iterates 70,000 times during training. The Pascal Voc2007 [14] dataset used in this experiment contains 20 objects, one for each category. In the experiment, the original method and our improved method are adopted respectively, and the VGG16 and ResNet101 basic training networks are combined. Table 3 shows the comparison of maps obtained by identifying different objects. The upper left corner of the box in the table represents the model's category for the target of the box area. The other columns are the traditional Faster R-CNN for VGG, our improved Faster R-CNN for VGG, the traditional Faster R-CNN for ResNet, and our improved Faster R-CNN for ResNet. The bold data in the table represents an improved score in the comparison results.

According to the data in the table, the average precision (AP) obtained using the original method and VGG16 as the underlying network is significantly lower than that using ResNet101. Using ResNet101, it is 9.06% higher than using VGG16. There are significant improvements in the correct rates for the four categories of birds, cups, chairs, and boats. The inconsistency in the promotion of each category is caused by the difference of the amount of data between the categories and the degree of difficulty. From the overall effect, the ResNet101 network has better capabilities in object detection than the VGG16 network. After improving Faster R-CNN, it is combined with the two network structures. Compared with the original method, the average rate of correctness is increased in multiple categories. The loss curve using different network structures during the experiment is shown in Figure 11.

Table 3. Comparison of Pascal Voc2007 scores by category (%)

Category	Original method + VGG	Our method + VGG	Original method + ResNet	Our method + ResNet
aero	66.44	**67.55**	71.37	70.14
bike	70.46	**71.41**	79.58	78.53
bird	58.26	**59.65**	70.57	69.74
boat	46.17	45.32	62.54	58.95
bottle	33.40	**39.86**	41.60	**41.79**
bus	69.84	**70.53**	78.19	**78.72**
car	63.30	**71.25**	71.60	**79.27**
cat	71.21	**78.39**	79.45	**79.69**
chair	33.61	**41.70**	46.78	**47.07**
cow	66.95	**72.65**	78.05	77.60
table	60.74	**65.70**	59.27	**66.32**
dog	70.29	**75.09**	79.66	79.29
horse	80.56	79.48	80.88	**81.27**
motorbike	61.71	**68.75**	70.08	**70.96**
person	62.11	**69.72**	70.68	**70.97**
plant	27.51	**28.43**	37.12	36.99
sheep	60.65	59.58	69.70	67.40
sofa	61.55	61.38	73.75	71.35
train	70.62	68.05	77.72	**77.92**
television	42.38	**51.47**	60.35	**67.35**
mAP (%)	58.89	**62.30**	67.95	**68.57**



(a) Using the VGG16 network (b) Using the ResNet101 network

Figure 11. Loss trend using different network structures

5. Conclusions

Based on the Faster R-CNN target detection framework, this paper designs an improved Faster R-CNN pedestrian detection method. The problem of identifying occlusions and small targets encountered by pedestrians under the general target detection framework is solved by adjusting the network structure of the extracted features and the RPN parameters of the candidate frames and the proportion of the generated anchor. The pedestrian detection model was generated by training with VGG16 and ResNet101 respectively, and the mAP values of 76.38 and 78.25 are obtained respectively. Finally, data migration verification was performed on the Pascal Voc2007 dataset. In practical problems, the detection efficiency is not up to the ideal level due to the issues of pedestrian mixing with the background, shielding, and illumination. In future research work, it is necessary to expand the training sample set and extract more information to improve the detection efficiency such as pedestrian attitude information, semantic information, and so on, which will be our future research direction in pedestrian detection.

References

1. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR'05*), Vol. 1, pp. 886-893, 2005
2. K. Schindler, N. Majer, S. Walk, and B. Schiele, "New Features and Insights for Pedestrian Detection," in *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 1030-1037, 2010
3. P. Dollar, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," in *Proceedings of the British Machine Vision Conference*, pp. 1-11, 2010
4. P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," in *Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 8, pp. 1-8, 2008
5. S. Tang, M. Andriluka, and B. Schiele, "Detection and Tracking of Occluded People," *International Journal of Computer Vision*, Vol. 110, No. 1, pp. 58-69, 2014
6. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 580-587, 2014
7. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, No. 9, pp. 1904-1916, 2015
8. R. Girshick, "Fast R-CNN," in *Proceedings of 2015 IEEE International Conference on Computer Vision* (*ICCV*), pp. 1440-1448, 2015
9. 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
10. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv: 1409.1556, 2014
11. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 770-778, 2016
12. W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," in *Proceedings of 30th Annual Conference on Neural Information Processing Systems* (*NIPS*), pp. 4905-4913, 2016
13. S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional Pose Machines," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 4724-4732, 2016
14. M. Everingham, S. M. A. Eslami, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, Vol. 111, No. 1, pp. 98-136, 2015

Shuang Liu received her Ph.D. in traffic information engineering and control from Dalian Maritime University. She is currently an associate professor at Dalian Minzu University.

Xing Cui is currently a postgraduate candidate at Dalian Minzu University.

Jiayi Li is currently a postgraduate candidate at Dalian Minzu University.

Hui Yang is currently a postgraduate candidate at Dalian Minzu University.

Niko Lukač obtained his Ph.D. in computer science in 2016 from Maribor University. He is currently a researcher in the faculty of Electrical Engineering and Computer Science at the University of Maribor.