Journal of Computers Vol. 31 No. 5, 2020, pp. 249-260
doi:10.3966/199115992020103105019

Research of Deep Learning in Pedestrian Detection

Chuan-wei Zhang*, Meng-yue Yang, Hong-jun Zeng, Bo Li

College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an Shaanxi 710054, China
zhangchw66@sina.com

Received 4 June 2019; Revised 9 September 2019; Accepted 21 November 2019

Abstract. The feature extraction of traditional manual design is complex and difficult to express the characteristics of pedestrians in complex scenes. To solve this problem, a deep learning network model is proposed. The model combines low-level features to form more abstract high-level to represent attribute categories or characteristics, from samples to extract more robust and better feature vectors. Because the network model has a deeper level, more training parameters, and fewer pedestrian data samples are labeled manually. A fine-tuning method is used to avoid over-fitting in the training process. Finally, experiments are verified on Caltech, INRIA and ETH pedestrian datasets. The data show that pedestrian detection algorithm of Faster R-CNN model has achieved 25%, 18% and 32% missed detection rates on Ped Faster RCNN-Visible respectively, which are higher than those on Ped Faster RCNN-Full. Experiments show that using occlusion can significantly reduce the performance of pedestrian detection. In the test phase, it can process a picture in an average of 0.31 seconds, which is 2.7 times faster than SA-Fast R-CNN and 20 times faster than R-CNN. It meets the real-time requirement in practical application.

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

1 Introduction

Pedestrian detection is an aspect of target detection. It is mainly used to separate pedestrians in pictures or video frames from complex or simple backgrounds to determine the position of pedestrians. Research on pedestrian detection technology began in the mid-1990s [1]. Until now, the technology has gradually become mature and real-time. Pedestrian detection technology can be applied to many aspects of real life, especially in the automobile auxiliary driving system plays an important role. With the gradual increase of vehicles, the road environment has become relatively complicated, and the safety of pedestrians has always been our focus. Whether they are domestic researchers or foreign researchers, they have done a lot of research on vehicle auxiliary systems using image processing technology, and constantly innovating and improving the vehicle's system model to improve the detection accuracy of vehicles on the road pedestrians, in order to protect the safety of pedestrians.

There are two main types of pedestrian detection for deep learning methods: One is pedestrian detection based on Faster RCNN algorithm [2]. This method introduces full convolution network, which improves the speed of feature extraction by nearly 200 times (about 10 ms). The detection accuracy is relatively high, but the speed is relatively slow. In 2015, Ren Shaoqing et al. [3] proposed a new improvement method based on Fast R-CNN [4]. This method is mainly about the generation of candidate boxes. Candidate boxes are extracted by introducing a full convolution Network of Region Proposal Network (RPN) [5]. The extraction speed of candidate boxes generated by RPN was nearly 200 times (about 10ms), so the algorithm framework for such target detection was called Faster R-CNN. In 2017, Xue Chao et al [6] improved the structure of Faster RCNN algorithm, and applied the methods of feature fusion, difficult case retraining and multi-scale pedestrian retraining to face detection. In 2018, Song Huansheng et al [7] solved the vehicle detection problem in complex environment by transforming Faster RCNN algorithm into two-class classification problem. It is because of the high detection ability of

* Corresponding Author

Faster RCNN that it has been widely used by later researchers. Another method is pedestrian detection based on YOLO (You only look once) algorithm [8]. The idea of this algorithm is to convert the target detection problem into a regression problem. The whole process of the algorithm is to input the original picture first, and then directly output the position and category of the object. Compared with Faster RCNN, the algorithm has the advantages of fast detection speed and almost tens of frames per second, but its accuracy is relatively low, and it is not sensitive to small target objects.

In short, there is a certain difference between pedestrian detection and general target detection, and pedestrian detection tends to solve real life problems. At the same time, the situation of missed detection and false detection often occurs in the process of pedestrian detection, which has great requirements on the accuracy and speed of detection. Therefore, this paper chooses VGGNet-16 network [9] for pedestrian feature extraction to better solve the difficult problems faced by pedestrian detection. A new ROI Pooling [10] has been added after the 13th layer convolutional layer to map input of different sizes to a fixed size feature vector, and then based on Faster R-CNN deep learning algorithm framework; network parameters are fine-tuned to make the deep learning network sensitive enough to pedestrian characteristics. As shown in Fig. 1. Four key algorithms of pedestrian detection are jointly learned in this paper: convolution layer output feature, candidate frame extraction, classifier and linear regression. Experiments are verified on Caltech, INRIA and ETH pedestrian datasets. The data show that pedestrian detection algorithm of Faster R-CNN model has achieved 25%, 18% and 32% missed detection rates on Ped Faster RCNN-Visible respectively, which are higher than those on Ped Faster RCNN-Full. Experiments show that using occlusion can significantly reduce the performance of pedestrian detection. The whole frame structure uses Fast R-CNN to extract features, classification and position refinement. Since the convolution layer of RPN and Fast R-CNN in the framework adopts the parameter sharing mechanism, the whole framework has end-to-end characteristics. In the test phase, it can process a picture in an average of 0.31 seconds, which is 2.7 times faster than SA-Fast R-CNN and 20 times faster than R-CNN. It meets the real-time requirement in practical application.

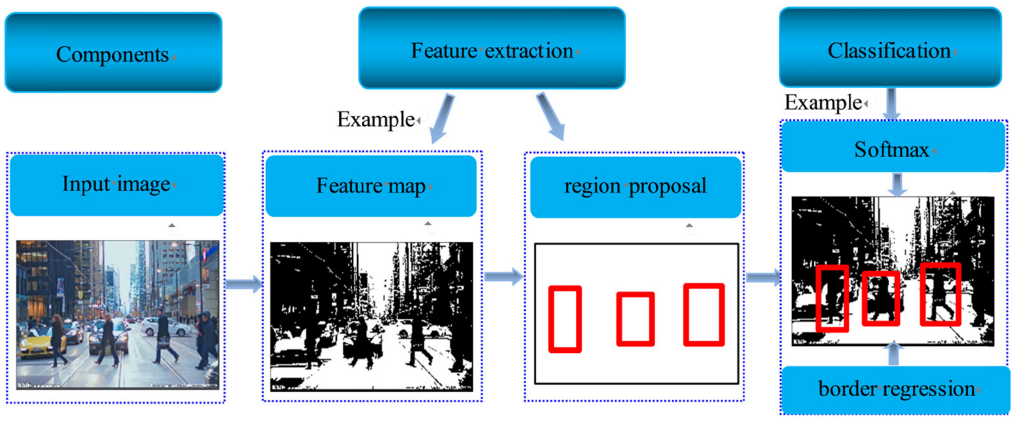


Fig. 1. Four key algorithms in pedestrian detection: convolution layer output features, candidate frame extraction, classifier and linear regression

2 Related Work

Pedestrian detection involves the selection of network structure and algorithm framework. This paper uses VGG-16 network [12] to extract features from pedestrians. A new aggregation layer (ROI Pooling) is added after the 13th layer convolutional layer to map different sized inputs to a fixed size eigenvector. Feature extraction and detection of pedestrians based on Faster R-CNN, Faster R-CNN deep learning model combines RPN network with Fast R-CNN network, The RPN network is used to generate high-quality candidate regions, and the Fast R-CNN network is used to extract relevant features in these candidate regions to determine whether there are pedestrians. Since both the RPN network and the Fast R-CNN network use the VGG-16 convolutional neural network structure, letting the convolutional layer of the two realize the sharing of network parameters can further accelerate the training speed of the model. Finally, two fully connected layers are connected after the ROI Pooling layer, and two Softmax classifiers are used to generate the final label box and the classification score.

The Faster R-CNN model consists of three basic frameworks. The first is the Region Proposal Network (RPN), which is used to generate candidate regions for each test picture. The second is a convolutional neural network, which is used to extract pedestrian features from candidate regions. The third is the binary Softmax classifier, which is used to classify the results. The overall structural framework is shown in Fig. 2.

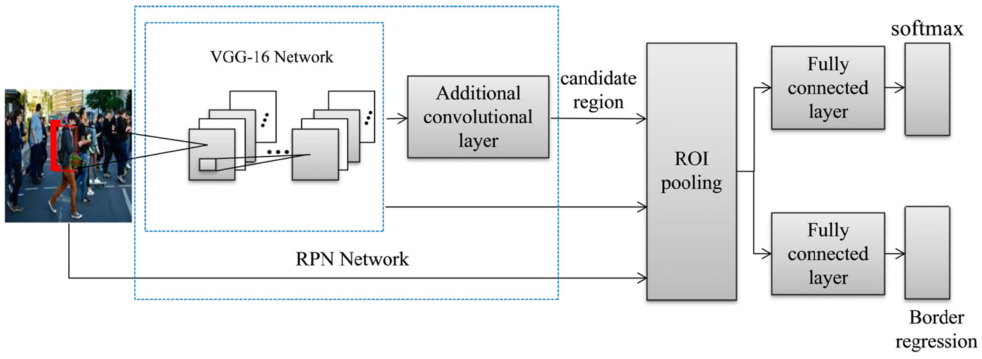


Fig. 2. Faster R-CNN pedestrian detection system block diagram. The Faster R-CNN model consists of three basic frameworks. The first is the Region Proposal Network (RPN), which is used to generate candidate regions for each test picture. The second is a convolutional neural network, which is used to extract pedestrian features from candidate regions. The third is the binary Softmax classifier, which is used to classify the results

3 Based on Improved leNet-5 CNN Pedestrian Detection Algorithm

3.1 RPN Network Structure

The RPN network uses pre-trained CNN to segment the image to directly generate candidate regions. It can receive images of any scale as input, and then output a number of candidate regions of the rectangle. Each region also has a confidence level of the corresponding target. Due to the flexibility of CNN, the RPN network can be fine-tuned to generate of the target category corresponding candidate regions. The VGG-16 model is used as CNN in this article. It can be found when analyzing the resource consumption of each layer of VGG-16 and the number of parameters. During the network training process, most of the memory and time are occupied by the previous convolutional layer. For example, the memory of the first convolutional layer occupies 224*224*64=3.2M, and the last two fully connected layer memory occupations are 4096. In contrast, most of the parameters exist in the last fully connected layer, and the parameters of the entire VGG-16 network are approximately 38M, while the first fully connected layer contains 100M parameters in the network training of pedestrian detection. This paper implements weight sharing between RPN network and Fast R-CNN network to improve the speed of network training and reduce repeated computation of the convolutional layer. The network model of RPN is shown in Fig. 3.

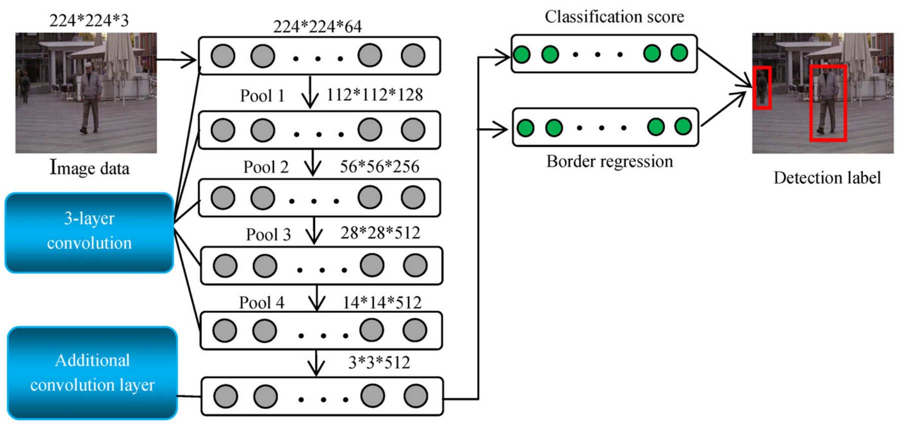


Fig. 3. RPN Network Model

The input of the VGG model is a picture of 224*224 sizes, and uses a 3 pixel fill. The convolution layer carry out 3*3 convolution with step length of 1;

2. The RPN network can accept images of any size as input and uses the convolution layer of the first 13 layers of VGG to convolve the input image;

3. Re-input the output of the 13th layer convolution layer onto an additional convolution layer. The convolution layer uses a 3*3 local receptive field to convolve the 13th layer convolution layer and then maps out a 512-dimensional feature vector;

4. Finally, they are input to two fully connected layers. The output is two Softmax classifiers, one is called the classification score layer, and the Softmax output category is 2 classes, which determines whether the pedestrian is included; The other is the border regression layer, which is used to fine-tune the resulting candidate regions.

3.2 Anchors Mechanism

In order to efficiently generate candidate regions using CNN, RPN network uses anchors mechanism [13]. As shown in Fig. 4, when the additional convolutional layer is convolved on the 13th layer convolutional layer, at the position of each sliding window, around the center point of the rectangular window, several candidate regions with multiple scales and multiple aspect ratios are predicted at the same time. These simultaneously predicted candidate regions are called anchor points, and the number of anchor points is usually 9, including 3 different scales (128, 256, 512) and 3 different length-width ratios (1:1, 1:2, 2:1), so each local receptive field corresponds to 9 anchors. We set the size of the 13th layer convolutional layer be $W_1 \times H_1$, which represents the width and height of the convolutional layer. The number of additional convolutional layer output neurons is A, which satisfies the following equation (1):

$$\begin{aligned} W_2 &= (W_1 - F + 2P)/S + 1 \\ H_2 &= (H_1 - F + 2P)/S + 1 \end{aligned} . \quad \textbf{(1)}$$

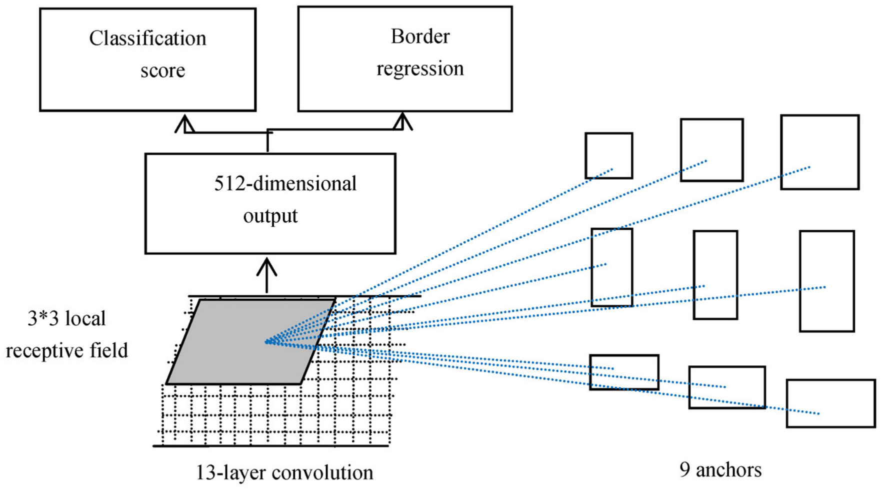


Fig. 4 Anchors Mechanism Diagram

They all output 512 dimensions. Since the local receptive field size is F=3, the pixel fill is P=1 and the sliding step size is S=1, the size of the additional convolutional layer output is as shown in equation (2):

$$\begin{aligned} W_2 &= (W_1 - 3 + 2)/1 + 1 = W_1 \\ H_2 &= (H_1 - 3 + 2)/1 + 1 = H_1 \end{aligned} . \quad \textbf{(2)}$$

It can be seen that the output of the additional convolutional layer is the same as the size of the 13th layer convolutional layer. So $W \times H \times 9$ anchors were obtained after 13th layer convolution layer of size $W \times H$ is convolved. Then the output of the additional convolutional layer is separately input into the classification score layer and the border regression layer. The border regression is used to indicate the offset between the candidate region coordinates and the ground truth. It produces the corresponding center position coordinates and the width and height of the rectangular border for each anchor point, so there are 9*4=36 coordinate values in border regression. The classification score layer is to determine whether the input image contains pedestrians and give the confidence score, so the output score is 2*9=18.

3.3 Border Regression

Since the candidate area generated by the RPN and the actual label position is relatively smaller, this paper uses the linear regression model to model this process. The objective function [14] can be expressed as shown in equation (3):

$$d_*(P) = w_* \phi(P). \tag{3}$$

$\phi(P)$ Represents the eigenvector of the additional convolutional layer output, w_* is a parameter that needs to be learned, expressed as w_x, w_y, w_w, w_h, $d_*(P)$ is the predicted translational amount and the amount of scaling. The amount of translation and scaling of the real need is as shown in equation (4)(5)(6)(7):

$$I_x = (G_x - p_x) / p_w \tag{4}$$

$$I_y = (G_y - p_y) / p_h \tag{5}$$

$$I_w = \log(G_w / p_w) \tag{6}$$

$$I_h = \log(G_h / p_h) \tag{7}$$

The loss function is shown in equation (8):

$$L = \sum_i^N (t_*^i - w_*^T \phi(P^i))^2 . \tag{8}$$

N represents the total number of pairs (G, P), and the optimization goal of the entire border regression is shown in equation (9):

$$w_* = \arg\min_{w_*} \sum_i^N (t_*^i - w_*^T \phi(P^i))^2 + \lambda \|w_*\|^2 . \tag{9}$$

λ is a regularized hyperparameter used to balance the loss function and the regularization penalty. In equation (9), the gradient descent method can be used to find the optimal solution. Then we use equation (3) to get $d_*(P)$, and further obtain the required amount of translation and scaling.

3.4 RPN loss Function

The RPN network assigns a binary class label to each anchor. Using Intersection over Union (IoU) to measure the degree of compliance between candidate regions and real annotations. The Intersection over Union of the two regions is defined as equation (10):

$$IoU = \frac{R_a \cap R_b}{R_a \cup R_b}. \tag{10}$$

Positive samples are defined as anchors of any real labeled maximum IoU. The negative sample is defined as an anchor that all true labeled IoU are less than 0.3.

The entire RPN network training is to solve a multi-task loss function, as shown in equation (11):

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, u) + \lambda \frac{1}{N_{reg}} \sum_i u L_{reg}(d_i, t_i^*). \tag{11}$$

i is the label of the anchors, p_i is the probability that an anchor predicts a pedestrian. u represents the label of the real label, if u is a positive sample, $u = 1$, otherwise $u = 0$. L_{cls} represents the loss function of the classification score, defined as equation (12):

$$L_{cls}(p_i, u) = -\log p_u. \quad \textbf{(12)}$$

$L_{reg}(d_i, t_i^*)$ is the border loss function. $uL_{reg}(d_i, t_i^*)$ means that the loss function is activated when the anchor label is positive (u=1). The above two loss functions are normalized by N_{cls} and N_{reg}. N_{cls} is the number of batch processing, taking $N_{cls} = 256$. λ is the number of anchors s, generally approximate 2400 is a hyperparameter to balance two loss functions, and to make them equal, taking $\lambda = 10$. The entire multitask loss function is solved through stochastic gradient descent.

4 Optimization Algorithm

This article uses the ImageNet data set to initialize the VGG-16 network, and then uses the pedestrian database Caltech to fine tune the network. First, set the output of the last layer of Softmax on the network to 2, which means whether there are pedestrians. Then let the RPN network share the VGG-16 convolutional layer with the Fast R-CNN network, and adopt the 4-step alternate training method:

(1) Step one: The VGG-16 is initialized with the ImageNet data set, and the weight is assigned to the RPN network;

(2) Step two: Use the Image Net trained VGG network and the candidate areas generated by the RPN to train the Fast R-CNN, the parameters of the two networks are not shared;

(3) Step three: Use the Fast R-CNN after the second training to retrain the RPN network, and set the learning rate of the shared convolution layer to 0, which is fixed shared convolutional layer. It only trains and updates RPN's unique network layer. At this time, the RPN network and the Fast R-CNN network have realized the sharing of the CNN convolutional layer.

(4) Step four: Fixed shared convolution layer. Use candidate regions generated by the RPN after the second training to train the Fast R-CNN, and fine tune the unique network layer of Fast R-CNN.

Through the above training of four steps, the Faster R-CNN network is sensitive enough to the pedestrian detection target, thereby, the function is realized which predict the candidate area and pedestrian detection in the network. 밭

In order to train the RPN, each candidate area is assigned a binary label, and the positive label can be assigned to two types of candidate areas: (1) candidate region overlapped with an IoU with a real target (GT) bounding box less than 0.5; (2) candidate areas overlapped with IoU with any GT boundary box greater than 0.5. A GT bounding box may assign positive labels to multiple candidate areas. Negative labels are only assigned to candidate regions with IoU ratios below 0.3 for all GT bounding boxes. Non-positive and non-negative candidate regions have no effect on the training target.

In the first stage of training, because the low-level convolution layer of CNN learns the general features of the input image, such as contour and texture, it does not change much and does not need to be adjusted, so the network parameters are updated from the 5th layer convolutional layer. In the second phase, only the layers unique of the RPN and Faster RCNN need to be trained, so only the 13th and the later layers are fine-tuned. The network is trained using the method of momentum update, the momentum μ is set to 0.9, the weight attenuation coefficient λ = 0.0005, and the initial learning rate α = 0.001.

5 Experimental Results and Analysis

In the simulation experiment, the evaluation criterion used is to calculate the number of false positive examples in each picture (False Positive Image, FPPI). The calculation method of FPPI is as follows. Each test image passes through the detection system. If there is no pedestrian in the image, and the system gives N test results, then the number of false positive cases adds N. If there are pedestrians in the image, judge whether the position of pedestrians and the IoU of each area marked by the network output are greater than the threshold (usually 0.5). If an IoU is below the threshold, the number of false positive cases is increased by 1. The final FPPI is the number of false positives divided by the total number of test pictures. By drawing the logarithmic space coordinate curves of FPPI and Miss Rate, the miss detection rates of 9 FPPI in 10^{-2} to 10^{0} logarithmic space with uniform spacing are averaged, the final logarithmic average miss detection rate is obtained as the standard, and various testing methods are evaluated.

In the simulation, the method of this paper is compared with some classic pedestrian detection methods, including VJ, HOG, Multi Ftr+Motion, HOGLBP, ACF-Caltech+, SCF+Alex Net and so on. The Faster RCNN proposed in this paper uses two different annotation data for training. The Faster RCNN proposed in this paper uses two different annotation data for training. Ped Faster RCNN-Full indicates that the model uses all the labels for training, while Ped Faster RCNN-Visible uses only the label of the visible part of pedestrians for training, without using occlusion related data.

5.1 Experimental Configuration

The software and hardware configuration of the machine are: the image and video processing platform is an ASUS desktop computer, the central processing unit (CPU) is Inter Core i7, and the frequency is 2.5 GHz. Graphics processor (GPU) NVIDIA GTX1080, 120GB solid state drive, memory size 4G. Use the Caffe framework in deep learning and refer to the concrete implementation of some of the hierarchy on the open source project.

5.2 Results of the Caltech Dataset

The Caltech 0-5 data set is used for model training and the 6-10 data set is used for testing. When training an RPN network, it contains 128 positive anchors (with pedestrians in the area) and 128 negative anchors (no pedestrians in the area). When training Fast R-CNN, there are 20 positive examples for candidate regions and 60 negative examples. Sample images are reduced by mean and normalization before entering the detection network, and then each time an image is input to the RPN network and two images are input to the Fast R-CNN.

As can be seen from Fig. 5. Ped Faster RCNN-Visible achieves a 25% miss rate when the false positive rate is 10%, which is much lower than the traditional statistical learning methods such as VJ, HOG, MultiFtr+ Motion and so on. At the same time, Ped Faster RCNN-Visible performance is better than Ped Faster RCNN-Full, indicating that the use of occlusion will significantly reduce the performance of pedestrian detection when training on the Caltech dataset.

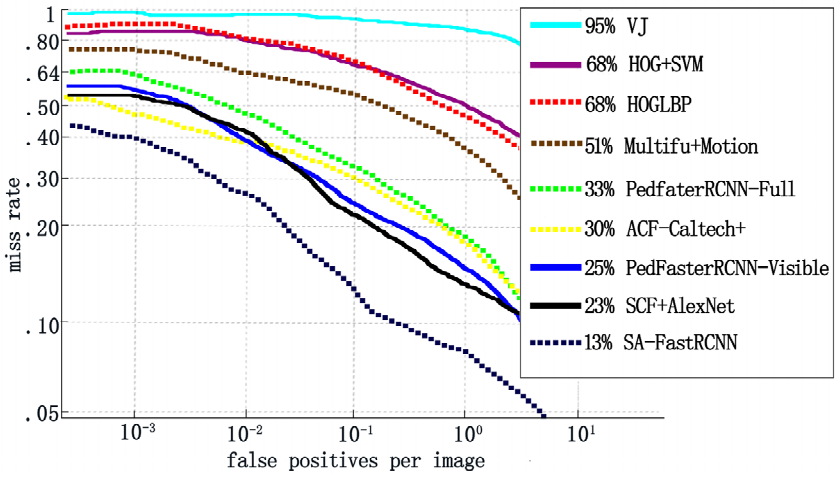


Fig. 5. Overall results on the Caltech-Test dataset

5.3 Results of the INRIA Dataset

This article also conducted experiments on the INRIA pedestrian dataset. However, due to the small number of pedestrian database samples (including 614 training samples and 218 test samples), it is easy to cause over-fitting when training the network, so we first extend the database before this training convolution neural network. This article will increase the pedestrian database of the literature [15]; the sample is similar to the INRIA data, and mainly taken on the campus of Fudan University, a total of 170. Therefore, the final training sample was 784 and the test sample was 218.

As can be seen from Fig. 6, when the false positive rate FPPI is 10%, Ped Faster RCNN-Full achieves 27% miss detection rate, which is about 12% lower than the traditional HOGLBP method. At the same time, Ped Faster RCNN-Visible achieved 18% missed detection rate and better performance than Ped Faster RCNN-Full. It shows that the use of occlusion can also significantly reduce performance of pedestrian detection when training on INRIA data sets, and it further demonstrates that the pedestrian detection system designed in this paper is superior to the traditional pedestrian learning based on machine learning.

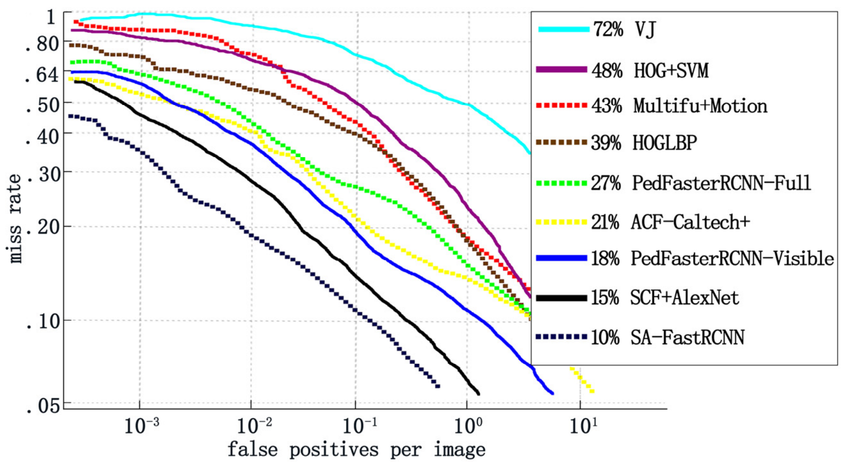


Fig. 6. Overall results on the INRIA-Test dataset

5.4 Results of the ETH Dataset

In addition, this article conducts experiments on the ETH dataset. The data of the ETH Pedestrian Database was captured by using a pair of binocular vehicle mounted camera in urban areas; the resolution is 640*480. The frame rate is 13-14fps, and the pedestrian's label information is given.

As you can see from Fig. 7, Ped Faster RCNN-Full achieves a 50% miss rate when the misjudgment rate FPPI is 10%; Ped Faster RCNN-Visible achieves a 32% miss rate and performs better than Ped Faster RCNN-Full. It shows that the performance of pedestrian detection can be significantly reduced by using occlusion when training on ETH data sets, so the occlusion-related data cannot be used as much as possible in training.

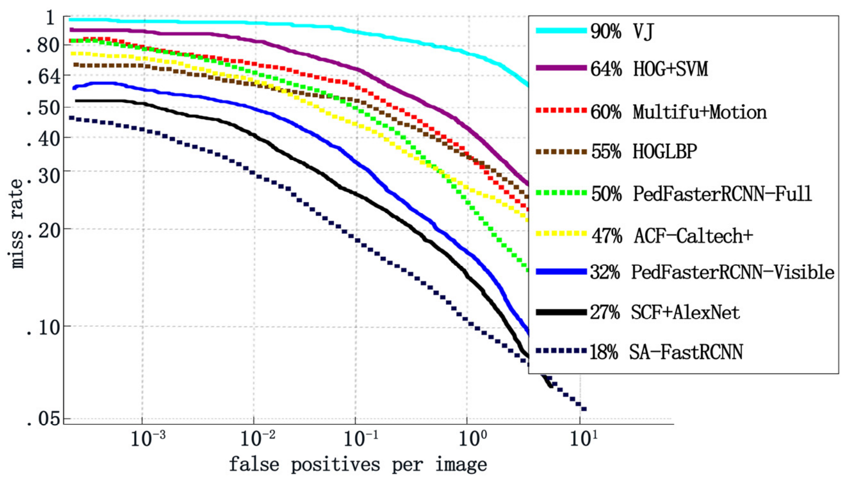
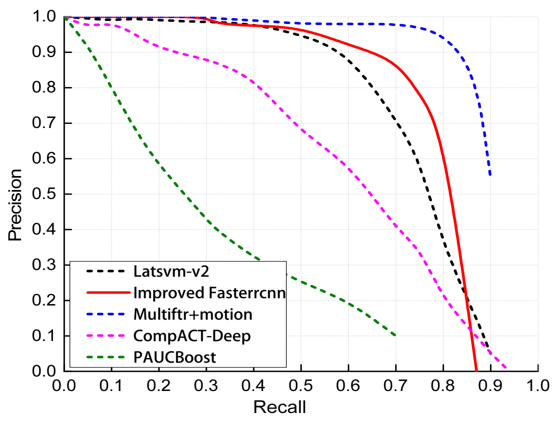


Fig. 7. Experimental results on the ETH dataset

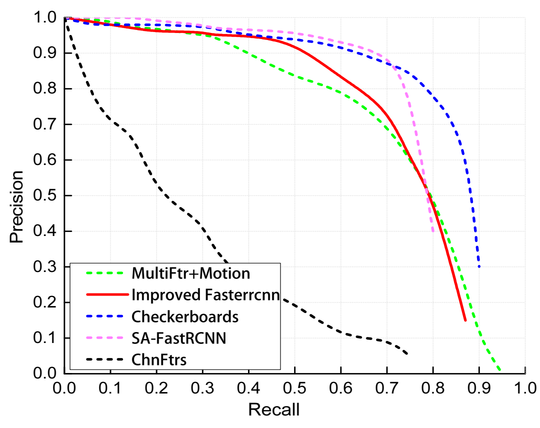
As can be seen from Fig. 5, Fig. 6 and Fig. 7, Using occlusion can significantly reduce the performance of pedestrian detection, so it is possible not to use occlusion related data during training.

Taking the Caltech dataset as an example, it can be seen from Fig. 5 that the best method for pedestrian detection, SA-fast RCNN, can achieve a 13% miss detection rate [15]. SA-fast RCNN is also a pedestrian detection method based on deep learning. Considering the influence of pedestrian size, different detection models are adopted according to different pedestrian heights. Although the deep learning method proposed in this paper has a gap of nearly 12% compared with SA-fastRCNN, the proposed Faster R-CNN method has scale invariance and near real-time processing speed, which can basically meet the actual needs.

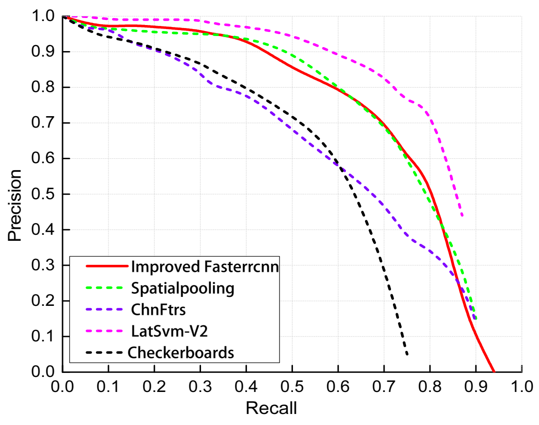
In order to verify the generalization ability of the model, the model is trained on the Caltech training set. It is tested on other datasets and compared with the better algorithm on the dataset, and the results are shown in Fig. 8. The graph represents the PR graph of each model on each dataset. in contrast to that ROC plot, the greater the value of the PR index, the better the performance. From the detection results, it can be seen that the detection effect of each algorithm on Caltech dataset is generally better. The difference between the results of this paper and PAUCBoost is not large, because most of the targets on the dataset are clear and easy to distinguish. Because the Daimler data set is a gray-scale image, the color information cannot be obtained, and the detection effect of the model is influenced, while the Multi Ftr Motion algorithm has achieved good performance after adding motion information, but the time complexity of calculating motion features is high. It takes about 200 seconds to process one frame of the image and cannot be applied to the application environment of the subject, so the motion features are not used herein. The scenes of large crowd aggregation were collected in USC and CVC data sets, that is, a large amount of pedestrian occlusion exist. However, the algorithm has a small gap compared with the Rand Forest and the Spatial Pooling algorithm. As a whole, the model in this paper can achieve good results in each dataset, and it can be seen that the model has a certain adaptability to different scenarios.



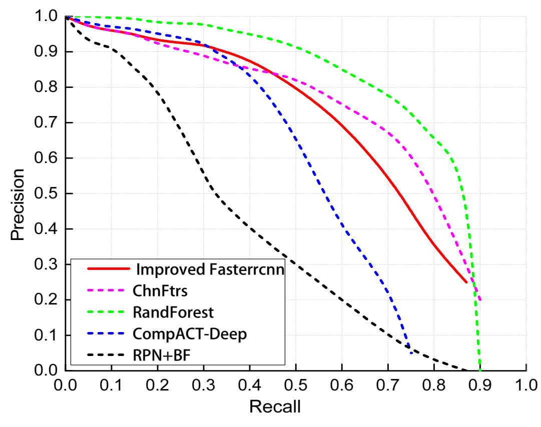
(a) Caltech



(b) Damiler



(c) USC



(d) CVC

Fig. 8. Model test results on different datasets

In addition, the effect of training iterations on detection results is also compared. Fig. 8 shows the variation of the loss function with the number of training iterations during training. It can be seen from Fig. 9 that the loss function converges when the number of training iterations reaches about 3,000, and subsequent retraining has little effect on the detection effect. Excessive training may also cause the entire neural network to be overly dependent on the current data set, causing over-fitting and thus affecting the detection effect.

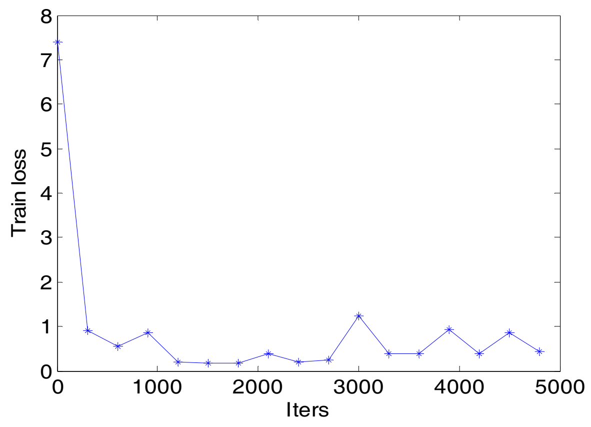


Fig. 9. Relation between loss function and iteration number

Table 1 shows the performance of the missing detection rate and the test time of each picture for different pedestrian detection algorithms with FPPI=0.1. In the Caltech test phase, it can process a picture in an average of 0.31 seconds, which is 0.27 time faster than SA-Fast R-CNN and 1.14 times faster than R-CNN. The result is shown in Table 1.

Table 1. Comparison of detection time and missing detection rate

Network name	Detection time (seconds/sheet)	Missing detection rate
Ours	0.31	25%
SA-Fast R-CNN	0.58	13%
R-CNN	1.45	15%

5.5 Pedestrian Detection Simulation in Real Scenes

The experimental video uses a high-definition camera to capture the natural scenes on campus. The scene is mainly the walkway in front of the teaching building. The trained model is input into a video for testing, and the detection results of some frames are shown in Fig. 10. It can be seen that the pedestrian target in all frames can be completely detected, the probability is very large, and the target boundary just happens to frame the target. In addition, because RPN uses three different scales, pedestrian targets with different distances in the image can also be marked with different sizes of boundary frames.

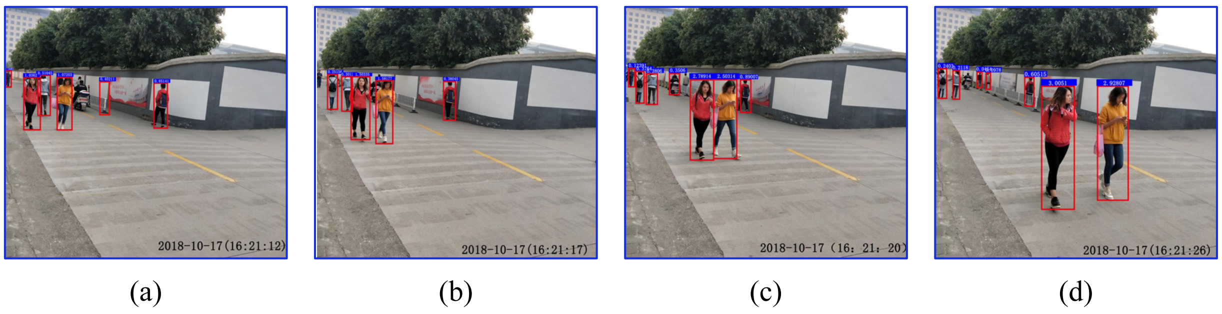


Fig. 10. Test results of training model

As can be seen from Fig. 9(a), When the Faster R-CNN model extracts features automatically; a column is misjudged as a human leg, so there is misdetection in the graph. It can also be observed from Fig. 10(a) to Fig. 10(d) that if the pedestrian is at a long distance, or he is riding a bicycle, the detection window fails to fully recognize the pedestrian. These problems need to be improved in the later stage.

6 Conclusion

This paper studies the application of deep learning network in pedestrian detection and constructs a complete Faster R-CNN network model. The model framework uses RPN instead of the traditional selective search method to generate candidate regions, and then inputs these candidate regions to the Fast R-CNN network for classification detection. In the classification detection network, the ROI sampling layer is still used to map the feature maps of the candidate regions from the original feature map. Both RPN and Fast R-CNN adopt multi-task loss update weights, and form an end-to-end framework by convolution layer parameter sharing mechanism, which reduces the training parameters. In training, it is performed in an alternating manner, so that the network weight can be optimized at the fastest learning speed. The results show that using occlusion can significantly reduce the performance of pedestrian detection. In the test phase, it can process a picture in an average of 0.31 seconds, which is 2.7 times faster than SA-Fast R-CNN and 20 times faster than R-CNN. The deep learning model effectively reduces the error rate of pedestrian detection, and the method has strong practical value.

Acknowledgements

This paper was supported by National Key R&D Program of China (2018YFC0808203) and Supported by Foundation of Shaanxi Key Laboratory of Integrated and Intelligent Navigation (SKLIIN-20180101). The authors would also like to express their deep gratitude for the support of the College of Mechanical Engineering, Xi’an University of Science and Technology.

References

[1] Y.-W. Xing, Research on pedestrian detection algorithm for driverless car vision system, [dissertation] Hebei, China: Yanshan University, 2018.

[2] L.-M. Gong, M.-H. Xu, D.-J. Liu. F.-Y. Zhang, Novel model of pedestrian detection based on Gaussian mixture model and HOG+SVM, Journal of Shanghai University 24(3)(2018) 341-351.

[3] M. Hemmati, M. Biglari-Abhari, S. Niar, S. Berber, Real-time multi-scale pedestrian detection for driver assistance systems, in: Proc. 2017 Design Automation Conference, 2017.

[4] J.-X. Zeng, X. Cheng, Pedestrian detection combined with single and couple pedestrian DPM models in traffic scene, Acta Electronica Sinica 44(11)(2016) 2668-2675.

[5] A. Verma, R. Hebbalaguppe, L. Vig, S. Kumar, E. Hassan, Pedestrian detection via mixture of CNN experts and thresholded aggregated channel features, in: Proc. 2015 International Conference on Computer Vision Workshop, 2015.

[6] C.-Y. Li, D. Song, R.-F. Tong, M. Tang, Illumination-aware faster R-CNN for robust multispectral pedestrian detection, Pattern Recognition 8(85)(2019) 161-171.

[7] C. Xiong, W.-W. Wang, Research on pedestrian detection based on DPM, Electronic Design Engineering 22(23)(2014) 172-173.

[8] Z.-Q. Zhao, H. Bian, D. Hu, W. Cheng, H. Glotin, Pedestrian detection based on fast R-CNN and batch normalization, in: Proc. 2017 International Conference on Intelligent Computing, 2017.

[9] F. Ye, C.-Y. Zhang, Y. Zhan, C. Ma, Real-time TV logo detection based on color and HOG features, in: Proc. 2013 International Symposium on Broadband Multimedia Systems and Broadcasting, 2013.

[10] W. Deng, B.-H. Liu, Deep learning-based pedestrian detection combined with semantics, Computer Systems & Applications 27(6)(2018) 165-170.

[11] X.-Y. Zeng, W.-L. Ouyang, M. Wang, X.-G. Wang, Deep learning of scene-specific classifier for pedestrian detection, in: Proc. 2014 European Conference on Computer Vision, 2014.

[12] A. Cuesta-Infante, F.-J. García, J.-J. Pantrigo, A.-S. Montemayor, Pedestrian detection with LeNet-like convolutional networks, Neural Computing & Applications 29(3)(2017) 1-7.

[13] A. Cardoso, M. Pires, R. Pinto, A. Cardoso, Implementation of keep-out-zones to protect sensitive sensor areas during backend processing in wafer level packaging technology, in: Proc. 2016 Electronic Components and Technology Conference, 2016.

[14] K.-M. He, X.-Y. Zhang, S.-Q Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Transactions on Pattern Analysis & Machine Intelligence 37(9)(2014) 1904-1916.

[15] Y. Li, J.M. Artés, B. Demir, S. Gokce, H.M. Mohammad, M. Alangari, J. Hihath, Detection and identification of genetic material via single-molecule conductance, Nature nanotechnology 13(12)(2018) 1167-1173.