International Journal of Advanced Robotic Systems
March-April 2019: 1–10
© The Author(s) 2019
DOI: 10.1177/1729881419829651
journals.sagepub.com/home/arx
SAGE

Pedestrian and cyclist detection based on deep neural network fast R-CNN

Kelong Wang[1,2] and Wei Zhou[3]

Abstract

In this article, a unified joint detection framework for pedestrian and cyclist is established to realize the joint detection of pedestrian and cyclist targets. Based on the target detection of fast regional convolution neural network, a deep neural network model suitable for pedestrian and cyclist detection is established. Experiments for poor detection results for small-sized targets and complex and changeable background environment; various network improvement schemes such as difficult case extraction, multilayer feature fusion, and multitarget candidate region input were designed to improve detection and to solve the problems of frequent false detections and missed detections in pedestrian and cyclist target detection. Results of experimental verification of the pedestrian and cyclist database established in Beijing's urban traffic environment showed that the proposed joint detection method for pedestrians and cyclists can realize the stable tracking of joint detection and clearly distinguish different target categories. Therefore, an important basis for the behavior decision of intelligent vehicles is provided.

Keywords

Intelligent driving, deep neural network, pedestrian detection, cyclist detection, fast R-CNN

Date received: 20 October 2018; accepted: 26 December 2018

Topic: Vision Systems
Topic Editor: Hongbo Gao
Associate Editor: Hongbo Gao

Introduction

The rapid development of intelligent driving technology has improved road traffic safety and urban traffic congestion. In particular, the protection of pedestrians and cyclists has attracted attention from governments, research institutes, and automobile companies. Effective detection and identification of pedestrians and cyclists is the prerequisite for their protection.

Pedestrian detection usually uses the sliding window method, which scales the image into different sizes and traverses the pedestrian area using a fixed-size window template. Dalal and Triggs used windows of different sizes to slide over the original image.[1] Felzenszwalb et al. scanned and correspond the low-resolution feature image through the root model and then matched the feature image twice the resolution of the root model through the component model.[2] Sermanet et al. combined the output of ConvNet multilayer convolution layer, global shape information, and local detail information to train the convolutional neural network model of pedestrian detection.[3] Similar to pedestrian detection, cyclist detection also uses the sliding window method. Cho et al. established a multiview cyclist detection model based on the deformable part model.[4] Li et al. designed a cyclist detection method based on improved histograms of oriented gradients (HOG)

[1] Graduate School of Chinese Academy of Social Sciences, Beijing, China
[2] Beijing Green Auto Technology Co., Ltd, Beijing, China
[3] CICC ALPHA (Beijing) Investment Fund Management Co., Ltd, Beijing, China

Corresponding author:
Kelong Wang, Graduate School of Chinese Academy of Social Sciences, Beijing Green Auto Technology Co., Ltd, Beijing 100101, China.
Email: kelongwang@126.com

features and support vector machine (SVM) classifier.[5] Yang et al. and Huang et al. proposed a multilayer cyclist detection method.[6,7] Tian and Lauer established the geometric relationship between the target size and image position based on the geometric constraints of the on-board camera; the scanning range of the image is restricted by cyclist size; and the cascade classifier of multi-view model is used to achieve cyclist target detection.[8] The difference between cyclist and pedestrian detection is that cyclists have different aspect ratios under different visual angles. A single fixed aspect ratio model is difficult to adapt to all cyclists. The use of multiple models will increase the amount of calculation and directly affect the practical application of detection algorithms.

The effect of target detection and classification increases when the target detection and classification model become complex. Some scholars have proposed common target detection and classification methods, including segmentation clustering and window scoring.[9] Segmentation clustering records the areas where detection targets may exist through image segmentation methods, including super pixel clustering, graph cut algorithm, and edge contours. Superpixel clustering generates target candidate regions by merging superpixel points, including the Selective Search,[10] Random Prime,[11] and Rankalankila.[12] Graph cutting solves the image segmentation problem through graph cutting algorithms,[13] including CPMC,[14] Endres,[15] and Rigor.[16] Edge contour combines the segmentation results by edge strength to generate target candidate regions.[17–19] Window scoring evaluates each candidate window to select a target candidate area. Among these methods, objectness[20] selects the initial candidate region by the prominent position in the image and then scores each candidate region by color, edge, position, and size. Rahtu et al. improved the scoring strategy and features based on the objectness method by generating a large number of initial candidate regions through independent/combined superpixels and multiple random sampling regions.[21,22] Bing identified the target candidate region through simple edge features and linear classifiers and consequently achieved rapid scoring, but the positioning accuracy needs to be improved.[23] Edge boxes select the target candidate region by fast sliding window method and use target edge estimation and individual tuning steps to improve positioning accuracy.[24] Feng et al. proposed a significance measurement method to generate candidate regions.[25] Zhang et al. and Li and Gao proposed a simple gradient feature concatenation SVM method to generate target candidate regions.[26,27] Among these methods, selective search, edge boxes, and region proposal network (RPN) methods provide good results but are only suitable for general object detection. The candidate region selection effect is not ideal for cyclist detection.

Challenges due to pedestrian posture, lighting, occlusion, and scale changes still exist in a real road environment. Compared with pedestrian detection, cyclist detection faces more challenges. Bicycle type and cyclist's clothes majorly change the appearance of the target, cyclist's posture changes the overall appearance of the target, and different observation angles change the aspect ratio of the target. Traditional pedestrian or cyclist detection methods usually treat these two targets separately, resulting in confusion of the detection results. The resolution of the traditional target detection model is limited, and effectively solving the above problems faced by pedestrians and cyclists is difficult.

A joint detection framework for pedestrians and cyclists based on deep neural network method is established in this study to solve the challenges faced by intelligent vehicles in the detection and identification of pedestrians and cyclists in complex driving environments. To solve frequent false detections and missed detections of pedestrians and cyclists, poor detection results of small-sized targets, and the complex and changeable background environment, this article presents the following main contributions: (1) a difficult case extraction method is designed based on the fast regional convolutional neural network, (2) a multilayer feature fusion method is designed, (3) an improved algorithm of depth network model is designed for multitarget candidate region input, and (4) a unified method of pedestrian and cyclist joint superscript detection and classification is constructed.

The remainder of this article is organized as follows. The second section introduces the target detection system architecture. Third section presents the pedestrian and cyclist detection methods. Fourth section describes the experiments of the algorithm and analysis results. Finally, fifth section presents some conclusive remarks.

Target detection system architecture

Target detection architecture

The target detection method based on fast region convolutional neural network is the most commonly used target detection framework and is characterized by excellent feature learning and classification ability of the deep convolution neural network model. The target candidate region is classified as the target and background to be detected, and the target recognition field has achieved a remarkable effect.

Fast region-based convolutional neural networks (R-CNN) proposes a multi-task simultaneous training model. The method also inputs candidate regions and the whole image extracted by the selective search method, obtains convolution feature maps through the multi-convolution layer and pooling layer, and extracts the feature vectors of fixed length using the pooling layer and the full connection layer of the region of interest (ROI). The feature vectors are encoded to two symbiotic output layers: one for estimating the target category and the other for predicting the target position. Unified training of the classification and positioning models is achieved without occupying large hard disk space, and fast training and detection are realized.

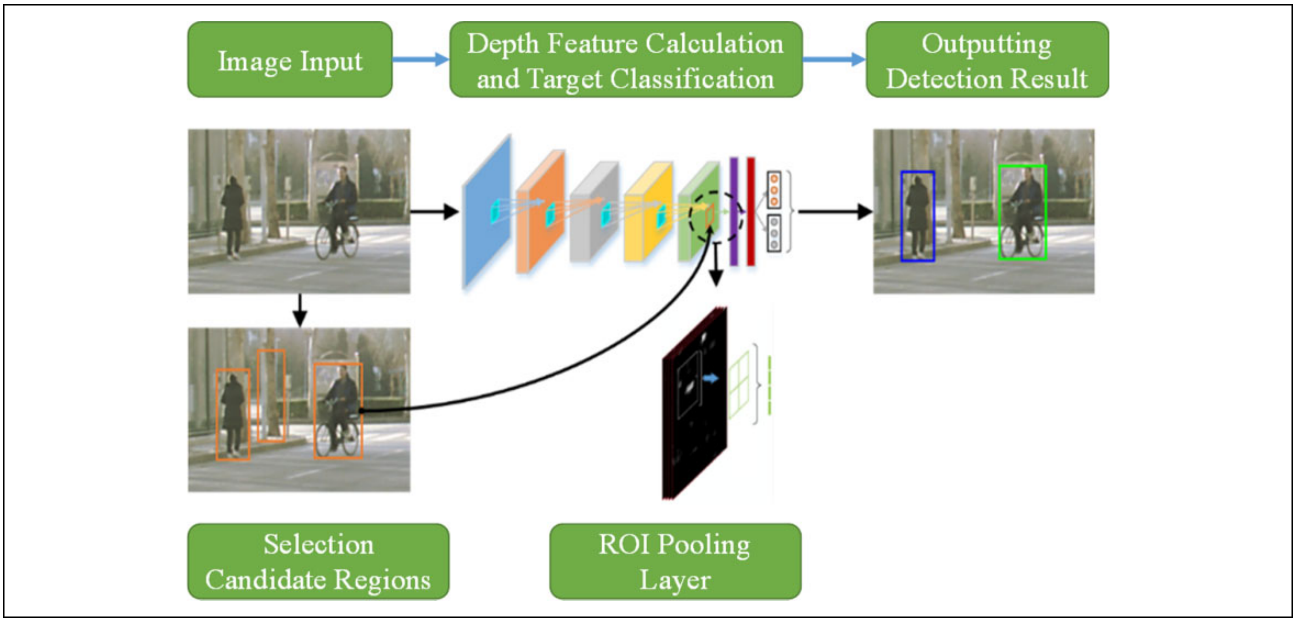
Image Input
Depth Feature Calculation and Target Classification
Outputting Detection Result
Selection Candidate Regions
ROI Pooling Layer

Figure 1. Target detection framework of fast R-CNN.

The target detection framework of fast R-CNN is shown in Figure 1.

The fast R-CNN region of interest pooling layer maps any ROI in the feature map (corresponding to the target candidate region in the original image) to a small feature map of fixed size H × W. Assuming that the size of the ROI is $h \times w$, the pooling layer of the ROI approximates the maximum value in each small h / H × w / W region as the mapping result of the region. This maximum value is the max pooling operation. Fast R-CNN defines a loss function that supports multi-tasking to achieve the goal of simultaneous multi-task training.

Fast R-CNN uses two symbiotic output layer networks, one of which is the estimated probability of each category $p = (p_0, \ldots, p_k)$, including the total of $k+1$ categories in the background. The other is the bounding-box regression offset corresponding to each category $t^k = (t^k_x, t^k_y, t^k_w, t^k_h)$, including the total of $k+1$ categories of background. For each real category u, the target with the regression quantity v of the corresponding real bounding box has a multi-task loss function as shown in the following equation

$$l(p, u, t^u, v) = l_{\text{cls}}(p, u) + \lambda \cdot 1\{u \geq 1\} \cdot l_{\text{reg}}(t^u, v) \tag{1}$$

Among them, the classification loss function is a logarithmic loss function, as shown in the following equation

$$l_{\text{cls}}(p, u) = -\log(p_u) \tag{2}$$

The second loss function is valid only when the real target category corresponding to the candidate region is not the background. The second loss function is shown in the following equation

$$l_{\text{reg}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t^u_i - v_i) \tag{3}$$

where $\text{smooth}_{L_1}(x)$ is shown below

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5\, x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

In equation (1), λ can adjust the weights of the classification and location loss functions. When t^u and v are normalized, $\lambda = 1$, good results are obtained.

Deep neural network architecture

The target categories to be detected are pedestrians and cyclists. The probability estimate of the corresponding fast R-CNN output layer includes the three categories (dimension 3) of pedestrians, cyclists, and background, and the bounding-box regression offset includes these three categories (dimension 12, where the background regression offset is zero). Fast R-CNN uses the target candidate region as the multi-example target candidate region selection method (MIOP), and the underlying network model used includes VGG models of different depths: VGG8, VGG11, and VGG16.[28] The VGG network structure diagram of different depths is shown in Figure 2, where the color filled squares represent the convolutional layers or fully connected layers with learning parameters.

Pedestrian and cyclist detection methods

Difficult case extraction network structure

Ren et al.[29] proposed an online difficult case extraction method to solve the problem of difficult case extraction in fast R-CNN target detection. This technique extracts difficult cases from many samples contained in each batch of training images instead of simply randomly selecting

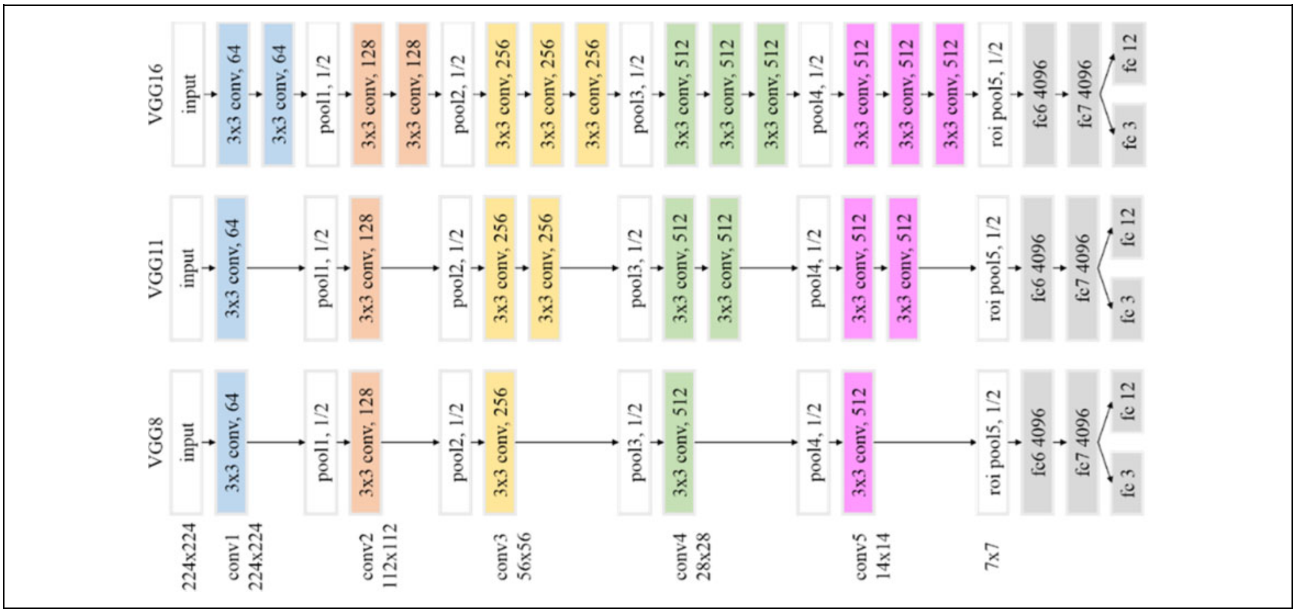

Figure 2. VGG network structure diagram of different depths.

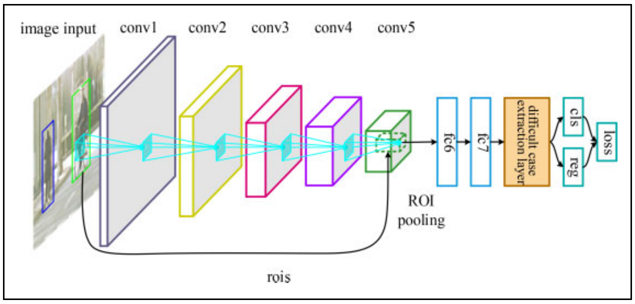

Figure 3. Structure diagram of the proposed training network for difficult case extraction.

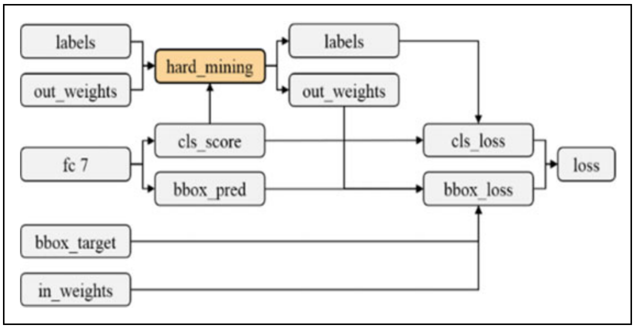

Figure 4. Network connection related to the difficult case extraction layer.

training samples. The selected difficult cases are immediately used for iterative network training, thereby changing the negative sample extraction conditions and random extraction methods in fast R-CNN to ensure that the final classification probability of samples can be used as the extraction basis. The results of this method are better than those of the traditional fast R-CNN method with a small increase in training time. Drawing on this idea, this study designs a corresponding difficult case extraction network structure for fast R-CNN target detection by replacing the two shared full connection layers and output layers with an original full connection layer and output layer. The structure diagram of the proposed training network for difficult case extraction is shown in Figure 3.

The network connection related to the difficult case extraction layer is shown in Figure 4. As shown in Figure 4, the input of the difficult case extraction layer consists of three parts, namely, the sample classification score (CLS_score), the real labels, and the outer weights (out _ weights) of the bounding-box regression. The sample classification score is the classification output result of the deep network to the sample, and the latter two are the input quantities of the network. The weight of the outer layer of the sample label is outputted, and bounding-box regression is conducted after the correction of the difficult sample extraction layer.

The difficult case extraction layer selects a certain proportion of samples as difficult cases based on the sample-classification scores. These selected difficult cases participate in the calculation of subsequent loss functions and updating of network parameters, and the remaining samples are ignored. Initially, a large number of samples (up to 2000) are randomly selected from each batch of training samples (minibatch) to be inputted into the network. Then, 10% of the samples (up to 200) are extracted from the difficult case extraction layer as the network loss function for the difficult case calculation to correct the network parameters. When selecting difficult samples, at most one-third of positive samples are selected, and the remaining ones are selected based on the sample scores. The label of unselected samples is set to one, the outer weight of bounding-box regression is set to zero, and the ignored samples are not included in calculating the classification loss function (CLS_LOSS) and bounding-box-regression loss function (BBOX_LOSS). The calculation of the classification loss function is shown in the following equation

$$L_{\mathrm{cls}} = \frac{1}{N}\sum_{i=1}^{N} 1\{u_i \geq 0\} \cdot l_{\mathrm{cls}}(p_i, u_i) \tag{5}$$

The calculation of the bounding-box regression loss function is shown in the following equation

$$L_{\mathrm{reg}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{out}_{\mathrm{weights}} \cdot \mathrm{smooth}_{L_1}\left(in_{\mathrm{weights}} \cdot (t_i - v_i)\right) \tag{6}$$

where N represents the number of samples inputted into the network per batch of training samples (minibatch) and is set to 2000; $l_{\mathrm{cls}}(p_i, u_i)$ is a logarithmic loss function, as shown in equation (2); *out_weights* and *in_weights* are the outer and inner weights of the bounding-box regression, respectively; and t_i and v_i represent the bounding-box-regression offset (BBOX _ PRED) and the corresponding true bounding-box-regression amount (BBOX _ TARGET), respectively, as shown in equation (3). The $\mathrm{smooth}_{L_1}(x)$ function is defined in equation (4). As shown in equations (5) and (6), when calculating the loss function of each batch of training samples, only the extracted difficult cases are considered and the unselected samples are ignored. The effective number of difficult cases is $N/10$, the calculated amount of loss function of the difficult cases is reduced by 10 times, and the gradient size is also reduced by 10 times when calculating the backward-propagation gradient. The backward-propagation weight update calculation is shown in the following equation

$$\begin{aligned} V_{t+1} &= \mu V_t - \alpha \Delta W_t \\ W_{t+1} &= W_t + V_{t+1} \end{aligned} \tag{7}$$

where W_t and W_{t+1} are the network weights at times t and $t+1$, respectively; V_t and V_{t+1} are the network weight updates at times t and $t+1$, respectively; ΔW_t is the backward-propagation weight gradient obtained at time t; μ is the inertia coefficient of the network weight updates at time t; and α is the learning rate. When the training difficultly extracts the network, the weight gradient of backward propagation is reduced by 10 times, and the learning rate α needs to be increased to obtain the appropriate training effect.

Multilayer feature fusion network structure

The structure of the multilayer feature fusion network is shown in Figure 5.

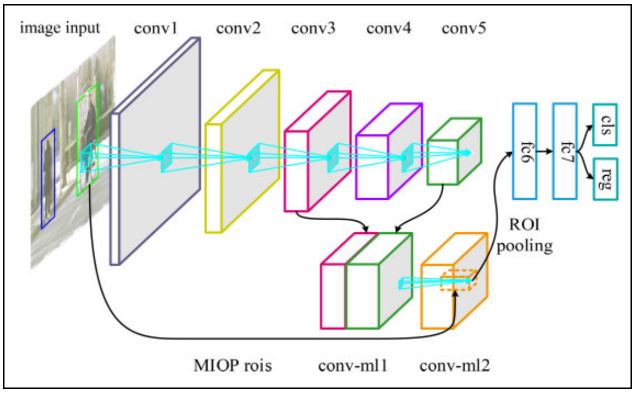

Figure 5. Structure of the multilayer feature fusion network.

For VGG 16 networks, we assume that the input image size of the network is 224 × 224, the feature size of the third convolution layer (CONV3-3) is 56 × 56, and the feature size of the fifth convolution layer (CONV5-3) is 14 × 14. To fuse two feature maps of different sizes, the third convolution layer is downsampled to 28 × 28 and the fifth convolution layer is upsampled to 28 × 28 so that the fusion of different feature maps can be realized. The third convolution layer is downsampled by the maximum pooling layer. Conversely, the fifth convolution layer is upsampled.

Considering the different amplitudes of activation values of convolution layers at different depths, linking up the feature maps sampled or reduced in dimensions at different layers results in information suppression or enhancement. Accordingly, the local response normalization operation proposed by Krizhevsky et al.[30] and Gao et al.[31] is used to smooth the activation values between different feature maps. The normalized activation value is shown in the following equation

$$b^i_{x,y} = a^i_{x,y} \Bigg/ \left(k + \alpha \sum_{j=\max(0,,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2\right)^{\beta} \tag{8}$$

where $a^i_{x,y}$ represents the activation value on an original feature map.

Candidate area selection network structure

To solve the selection of candidate regions, the RPN network structure is designed, as shown in Figure 6. The classification result and regression offset of each target candidate region are used to obtain the candidate regions that may contain targets. The design of the reference bounding box relies on the image input of a single size to extract target candidate regions with different aspect ratios and sizes. Based on the characteristics of pedestrian and cyclist targets, this article designs a reference bounding box with three aspect ratios (1: 1, 2: 1, and 3: 1) and five dimensions (32 × 32, 64 × 64, 128 × 128, 256 × 256, and 512 × 512). According to the set reference bounding-box parameters, 15 reference bounding boxes can be generated for each position in the feature map.

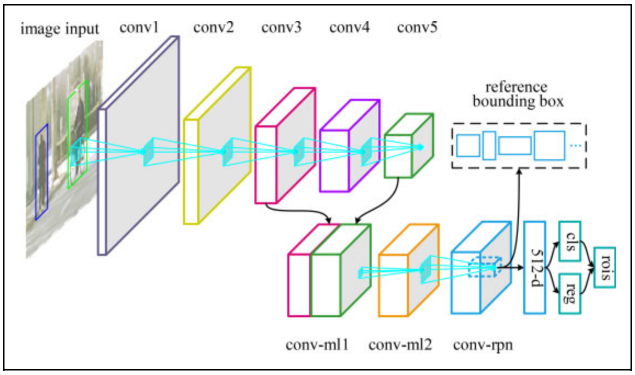

Figure 6. RPN network structure.

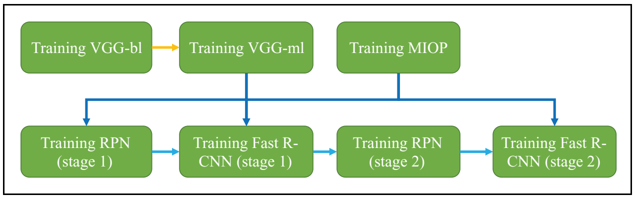

Figure 7. Overall network architecture for pedestrian and cyclist detection.

The RPN training network includes two symbiotic output layers, one representing the estimated probability p_i of positive samples and the other representing the bounding-box-regression offset t_i. The ith real target category is marked as p_i^* (positive sample is one, negative sample is zero), and the true bounding-box offset is marked as t_i^* (corresponding to the four parameters x, y, w, h). RPN's multitask loss function is shown in the following equation

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i l_{\text{cls}}(p_i, p_i^*) + \frac{1}{N_{\text{reg}}} \sum_i p_i^* l_{\text{reg}}(t_i, t_i^*) \tag{9}$$

where $l_{\text{cls}}(\cdot)$ is a logarithmic loss function, as shown in equation (2); $l_{\text{reg}}(\cdot)$ is a smooth L1 loss function, as shown in equation (3); λ is used to adjust the weights of the classification loss function and the location loss function; and N_{cls} and N_{reg} are the batch training and scale of all target candidate regions, respectively.

Overall network structure

Considering difficult case extraction, multilayer feature fusion, multicandidate region input, RPN, and fast R-CNN convolution layer sharing, we design the overall network architecture for pedestrian and cyclist detection, as shown in Figure 7.

As shown in Figure 7, the overall network model is obtained through training to achieve effective detection of pedestrian and cyclist targets. The method comprises the following steps: (1) a group of target candidate regions is extracted according to the MIOP, (2) another group of target candidate regions is extracted through RPN, and (3) the two groups of target candidate regions are classified and optimally positioned through fast R-CNN, thereby realizing target detection. The last two steps realize the network parameter sharing of the convolution layer, and the network structure diagram is shown in Figure 8.

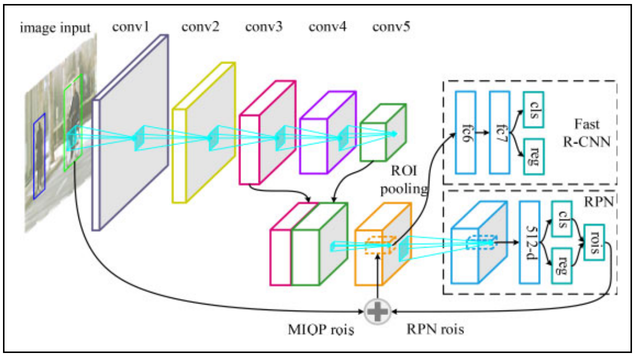

Figure 8. Network structure of parameter sharing of the last two steps of the convolution layer.

Experiment and result analysis

Model training

To verify the effect of pedestrian and cyclist target detection network, a large number of network models are trained, and their training network models are shown in Figure 9. "bl" represents the basic network and does not consider any improvement methods, "hm" means considering difficult case extraction, "ml" means considering multilayer feature fusion, "ml-hm" means that both multilayer feature fusion and difficult case extraction are considered, "faster" means faster R-CNN, "final" means the final overall network, "iter" indicates the training period of the network, and "lr" indicates the learning rate of the network. The interconnected upper- and lower-layer network model in Figure 9 shows that the upper-layer network model is trained based on the lower-layer network model.

Experimental results and analysis

To verify the effect of the deep neural network model for pedestrian and cyclist detection, the VRU database was used for experimental verification, and PR curve representation and average accuracy were used for evaluation. As shown in Figures 10 and 11, when counting the test results of pedestrians and cyclists, they are evaluated and verified in the verification sets of different difficulty levels. The influence of interference categories is ignored.

We compare the detection results of pedestrians and cyclists in different basic network models. As shown in Figures 10 and 11, VGG8-bl has better detection effect and shorter average detection time than VGG11-bl and VGG16-bl, but the cyclist's detection effect is slightly

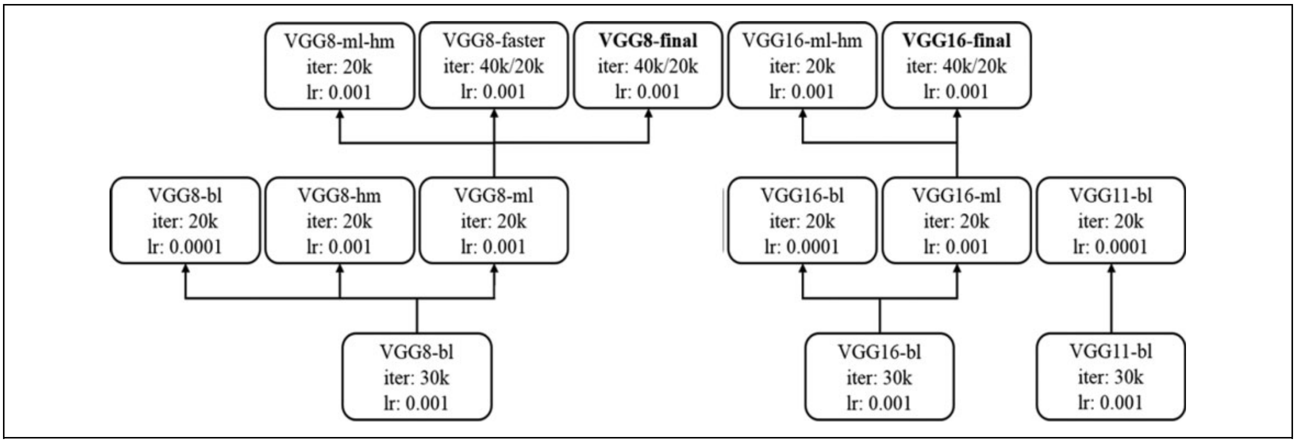

Figure 9. Training network models.

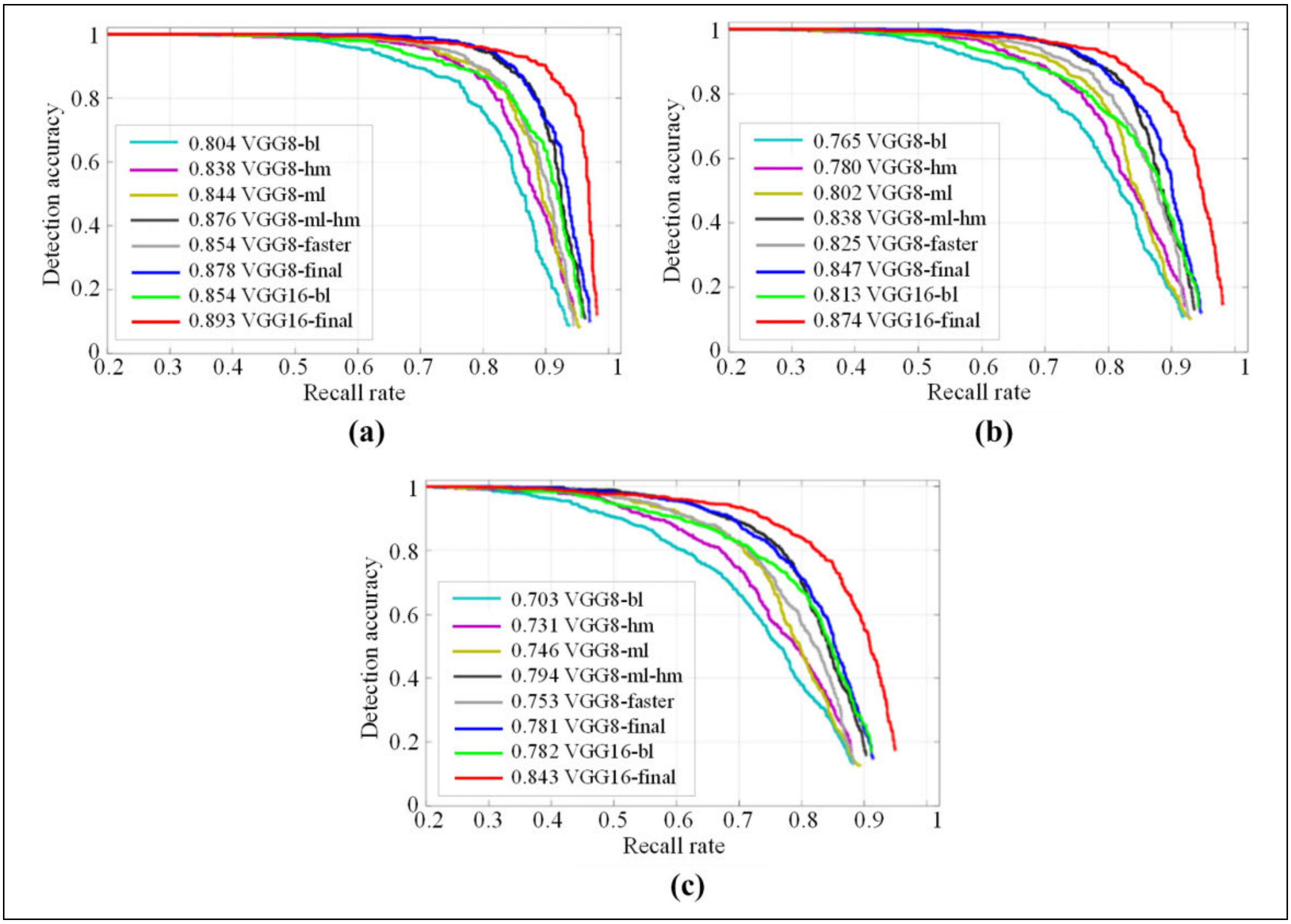

Figure 10. Pedestrian detection results: (a) simple difficulty-level verification set, (b) medium difficulty-level verification set, and (c) high difficulty-level verification set.

poorer and the average detection time is slightly longer. Thus, VGG8 is selected as the basic network model for verifying the network improvement scheme to clearly and intuitively compare the detection results of different network models on different difficulty-level verification sets, as shown in Figures 10 and 11.

As shown in Figures 10 and 11, the average accuracy of VGG8-final on VGG8-ml-hm, VGG8-final in different difficulty-level verification sets is slightly high, whereas the average accuracy of cyclist detection is basically flat, although the relative VGG8-faster, VGG8-final advantage is obvious. The average accuracies of pedestrian detection

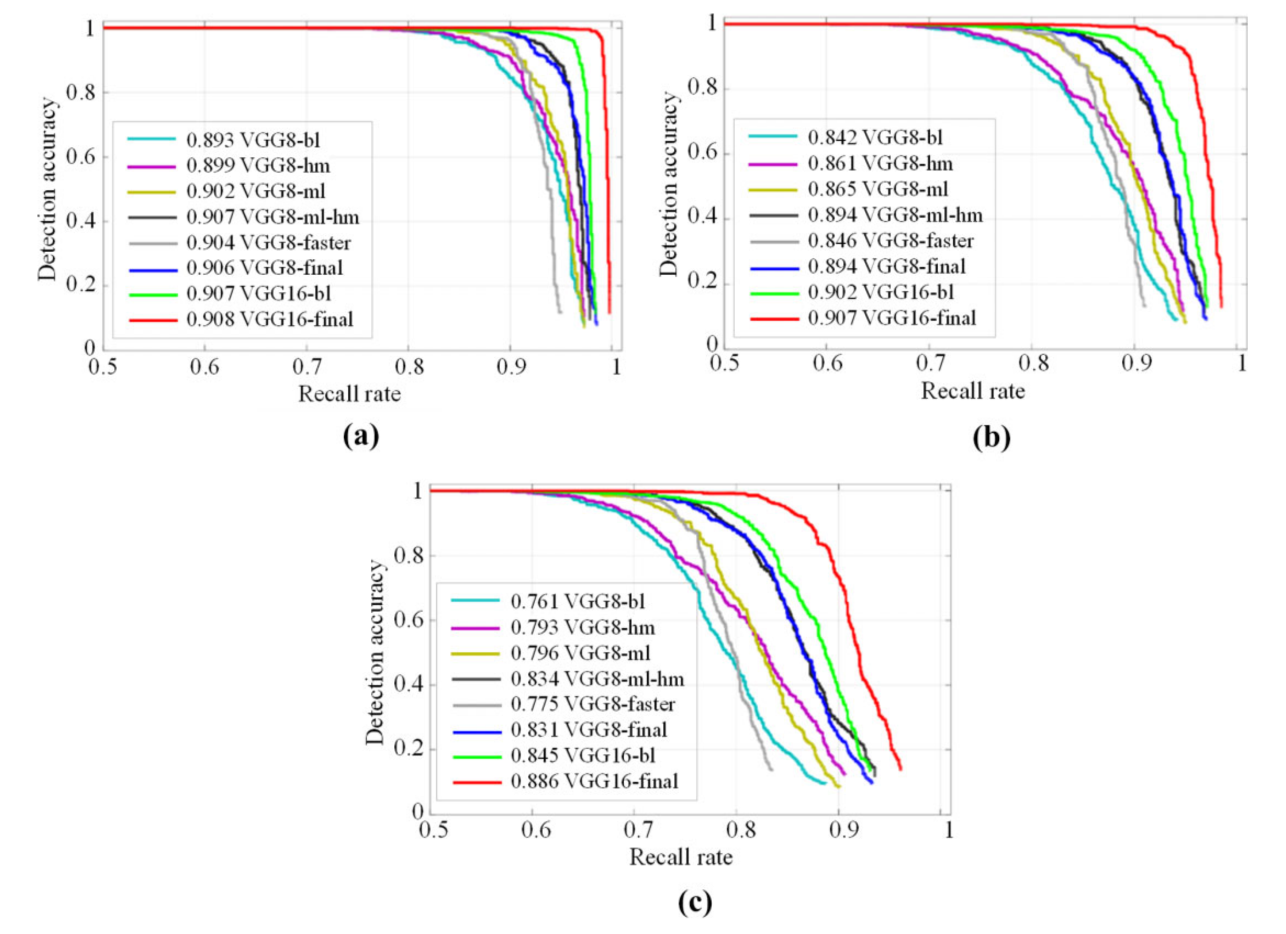

Figure 11. Cyclist detection results: (a) simple difficulty-level verification set, (b) medium difficulty-level verification set, and (c) high difficulty-level verification set.

for simple, medium, and high difficulty levels were 2.4%, 2.2%, and 4.7%, respectively, and the average accuracies of cyclists were 0.2%, 4.8%, and 5.8%, respectively. Compared with VGG16-ml-hm, VGG16-final, the average accuracy of pedestrian detection in different difficulty-level verification sets, is 1.7%, 4.1%, and 6.6%, and the average accuracy of cyclist detection is basically flat. Experimental results indicate that the VGG8/16-final detection effect is better than using a method alone to achieve the advantages of pedestrian and cyclist detection.

As shown in Figures 10 and 11, the VGG8-final detection effect significantly affects the basic network model VGG8-BL. The average accuracies of pedestrian detection for simple, medium, and high difficulty levels are 7.4%, 8.2%, and 9.7%, respectively, and the average accuracies of cyclist detection are 1.3%, 5.2%, and 6.9%, respectively. The VGG16-final detection effect is significantly improved compared with the basic network model VGG16-BL, especially the pedestrian detection effect, in different difficulty-level verification sets. On the average, the pedestrian detection accuracies were 3.9%, 6.1%, and 6.1%, respectively, and the average rider detection accuracies were 0.1%, 0.7%, and 4%, respectively, for simple, medium, and high difficulty levels. The validity of the proposed network model is further verified. Compared with the overall network model VGG16-final and VGG8-final, the detection effect of VGG16-final is better than VGG8-final, and the average accuracies of pedestrian detection on different difficulty-level verification sets are 1.5%, 2.7%, and 4.3%, respectively. The average accuracies of cyclist detection are 0.2%, 1.5%, and 5.6%, respectively, indicating that a deeper network helps improve the results of final target detection.

Conclusion and future work

In view of the existing deep-learning methods used for pedestrians and cyclists to detect deficiencies, based on the fast R-CNN target detection framework, we focus on the following to design a comprehensive difficult sample extraction method and multilayer feature fusion: pedestrian and cyclist target error detection; frequent, small-size targets; and difficult to detect, changeable, and complex environment background. Many improved network structure models such as multitarget candidate region input greatly improve the detection effect of pedestrian and cyclist targets. In the course of network training, the use of difficult sample extraction instead of random sampling to select negative samples effectively enhances the pedestrian and cyclist target detection effect, thereby reducing the complex driving road environment that causes false detection of pedestrian and cyclist targets. A convolution feature map with different depths can synthesize local and global features, obtain stronger feature information, improve pedestrian and rider target detection effect, and enhance the effect of small-size pedestrian and cyclist targets. By combining the inputs of two target candidate regions to compensate for the defect of the single target candidate region, the complementary advantages of the MIOP and RPN methods are realized. Thus, the detection effects of pedestrian and cyclist targets are further improved, thereby leading to a reduction in missing pedestrians and cyclist targets.

In future work, the pedestrian and cyclist target detection method based on deep learning proposed in this article will be applied to the real environment test of intelligent driven vehicles to improve the target detection rate of pedestrians and cyclists. The utility of the proposed method will be verified.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: National Key Research and Development Program of China, (Grant / Award Number: ‘2017YFB0102603’).

ORCID iD

Kelong Wang https://orcid.org/0000-0002-5285-3940

References

1. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, 20–26 June 2005, pp. 886–893. IEEE.
2. Felzenszwalb PF, Girshick RB, McAllester D, et al. Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal* 2010; 32(9): 1627–1645.
3. Sermanet P, Kavukcuoglu K, Chintala S, et al. Pedestrian detection with unsupervised multi-stage feature learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland Oregon, 25–27 June 2013, pp. 3626–3633. IEEE.
4. Cho H, Rybski PE, and Zhang W. Vision-based bicyclist detection and tracking for intelligent vehicles. In: *Proceedings of the IEEE Conference on Intelligent Vehicles Symposium (IV)*, San Diego, 21–24 June 2010, pp. 454–461. IEEE.
5. Li T, Cao X, and Xu Y. An effective crossing cyclist detection on a moving vehicle. In: *Proceedings of the IEEE 8th World Congress on Intelligent Control and Automation (WCICA)*, Jinan, China, 7–10 July 2010, pp. 368–372. IEEE.
6. Yang K, Liu C, Zheng JY, et al. Bicyclist detection in large scale naturalistic driving video. In: *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 8–11 October 2014, pp. 1638–1643. IEEE.
7. Huang GB, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern B (Cybern)* 2012; 42(2): 513–529.
8. Tian W and Lauer M. Fast cyclist detection by cascaded detector and geometric constraint. In: *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, Spain, 15–18 September 2015, pp. 1286–1291.
9. Hosang J, Benenson R, Dollár P, et al. What makes for effective detection proposals? *IEEE Trans Pattern Anal* 2016; 38(4): 814–830.
10. Uijlings JRR, van de Sande KEA, Gevers T, et al. Selective search for object recognition. *Int J Comput Vision* 2013; 104(2): 154–171.
11. Manen S, Guillaumin M, and Van Gool L. Prime object proposals with randomized Prim’s algorithm. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, 3–6 December 2013, pp. 2536–2543. IEEE.
12. Rantalankila P, Kannala J, and Rahtu E. Generating object segmentation proposals using global and local search. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 24–27 June 2014, pp. 2417–2424. IEEE.
13. Felzenszwalb PF and Huttenlocher DP. Efficient graph-based image segmentation. *Int J Comput Vision* 2004; 59(2): 167–181.
14. Carreira J and Sminchisescu C. CPMC: automatic object segmentation using constrained parametric min-cuts. *IEEE Trans Pattern Anal* 2012; 34(7): 1312–1328.
15. Endres I and Hoiem D. Category-independent object proposals with diverse ranking. *IEEE Trans Pattern Anal* 2014; 36(2): 222–234.
16. Humayun A, Li F, and Rehg JM. RIGOR: reusing inference in graph cuts for generating object regions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 24–27 June 2014, pp. 336–343. IEEE.
17. Gao HB, Cheng B, Wang JQ, et al. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Trans Ind Inform* 2018; 14(9): 4224–4231.
18. Arbeláez P, Pont-Tuset J, Barron JT, et al. Multiscale combinatorial grouping. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 24–27 June 2014, pp. 328–335. IEEE.
19. Krähenbühl P and Koltun V. Geodesic object proposals. In: *European Conference on Computer Vision*, Zurich,

Switzerland, 6–12 September 2014, pp. 725–739. Springer International Publishing.

20. Alexe B, Deselaers T, and Ferrari V. Measuring the objectness of image windows. *IEEE Trans Pattern Anal* 2012; 34(11): 2189–2202.
21. Rahtu E, Kannala J, and Blaschko M. Learning a category independent object detection cascade. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 6–13 November 2011, pp. 1052–1059. IEEE.
22. Xie GT, Gao HB, Wang JQ, et al. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Trans Ind Electron* 2017; 56(7): 5999–6008.
23. Cheng MM, Zhang Z, Lin WY, et al. BING: binarized normed gradients for objectness estimation at 300fps. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 24–27 June 2014, pp. 3286–3293. IEEE.
24. Zitnick CL and Dollár P. Edge boxes: locating object proposals from edges. In: *European Conference on Computer Vision*, Zurich, Switzerland, 6–12 September 2014, pp. 391–405. Springer International Publishing.
25. Feng J, Wei Y, Tao L, et al. Salient object detection by composition. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 6–13 November 2011, pp. 1028–1035. IEEE.
26. Zhang Z, Warrell J, and Torr PHS. Proposal generation for object detection using cascaded ranking SVMs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, 21–23 June 2011, pp. 1497–1504. IEEE.
27. Li DY and Gao HB. A hardware platform framework for an intelligent vehicle based on a driving brain. *Engineering* 2018; 4(2018): 464–470.
28. Erhan D, Szegedy C, Toshev A, et al. Scalable object detection using deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 24–27 June 2014, pp. 2147–2154. IEEE.
29. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *Adv Neur In* 2015; 39(6): 91–99.
30. Krizhevsky A, Sutskever I, and Hinton G E. Imagenet classification with deep convolutional neural networks. *Adv Neur In* 2012; 25(2): 1097–1105.
31. Gao HB, Zhang TL, Liu YC, et al. Research of intelligent vehicle variable granularity evaluation based on cloud model. *Acta Elect Sinica* 2016; 44(2): 365–374.