FSSPD: Fast Single Stage Pedestrian Detector for Autonomous Driving

Article · February 2019

DOI: 10.12783/dteees/iceee2018/27819

3 authors, including:

Ying Shi
Wuhan University of Technology
16 PUBLICATIONS **131** CITATIONS
SEE PROFILE

Changjun Xie
Wuhan University of Technology
74 PUBLICATIONS **357** CITATIONS
SEE PROFILE

International Conference on Electric and Intelligent Vehicles
ICEIV 2018, Nov 21-25, 2018, Melbourne, Australia
Paper ID: ICEIV2018-xxx

FSSPD: Fast Single Stage Pedestrian Detector for Autonomous Driving

Jia-qi Luo[1], Ying Shi[1,*], Chang-jun Xie[1,*]

1 Wuhan University of Technology, Wuhan, China

* Corresponding author: Ying Shi (a_laly@163.com), Chang-jun Xie (jackxie@whut.edu.cn)

Abstract

For low accuracy of the anchor-based pedestrian detectors in the case of small and high-density pedestrian, a Fast Single Stage Pedestrian Detector (FSSPD) is presented, which has three merits. Firstly, a single stage anchor-based neural network is designed on the base of the modified Darknet-19 instead of the widely used VGG-16, largely reducing inference time during pedestrian detection. Secondly, Darknet-19 usually adapted in YOLOv2 is modified by using a scale invariant network structure with multi detection modules to detect small pedestrians. Thirdly, a dense anchor strategy is proposed to deal with high mistake rate and miss rate of high-density pedestrian. As a result, the proposed network structure and strategies are proved effective for small and high-density pedestrian detection, and the multi-detection-module strategy works well for small pedestrian with significantly improving the average precision on the evaluation set of KITTI dataset by 15.8%, 16.5% and 17% in easy, moderate and hard level, respectively. And our approach's accuracy outperforms the popular YOLOv2 approach on the test set of KITTI dataset by 39.86%, 30.96%, and 27.25% in the easy, moderate and hard difficulty level, respectively, at the speed of 14.3fps in near real time.

Keywords: pedestrian detection, autonomous driving, deep learning, fast single stage detector.

1. Introduction

Pedestrian detection widely used in autonomous driving is a challenging task due to the changes of posture, skin colors, age, illumination and light from different directions, especially the disturbance from small and high-density pedestrian [1].

Many approaches for pedestrian detection have been proposed via the use of the hand-crafted features and the optimal classifiers [2-7]. Most of them have gotten good results in simple cases, such as no occluded pedestrian, with performances relying heavily on the hand-crafted features and do not work well for complex situations.

Recently, thanks to the development of hardware and the breakthroughs of deep learning, many works based on convolutional neural networks (CNN) have greatly improved the accuracy of pedestrian detection [8-11]. Most of them regard pedestrian as a special object and pedestrian detection algorithm can be inherited from that of object detection. [12] showed that a well-performed CNN, Fast-RCNN, applied in object detection also behaved excellently in pedestrian detection. Now the proposed CNN-based object detections are mostly based on two stage proposal-classification detector [13, 14] or single stage detector [15-17]. The two stage detector, such as Fast-RCNN [14], includes two stages, that is, region proposing and object classification. While single stage detector, such as SSD [16] and YOLOv2 [17], uses an end-to-end forward neural network architecture to get much higher speed without the accuracy drops largely.

For single stage CNNs, it is important to choose a good underlying network, among which VGG16 [18] is the most popular one and ResNet [19] or Inceptionv4 [20] also can be used. Most of them get high accuracy but with high computation. Among most state-of-the-art single stage CNNs, YOLOv2 shows a good speed performance on object detection because of the use of a new architecture, Darknet19 network, but a poor accuracy performance in the case of dense pedestrians for its large stride size.

Small or occluded object often exists in the case of high-density pedestrian and its detection is still a bottleneck of this domain. [21] adopted a scale invariant network structure from SSH in face detection and got a good detection on small faces by the use of a scale invariant network structure consisting of multi detection modules. Besides network structure, accurate parameter determination for default anchor boxes in training phase is also significant for pedestrian detection. Many methods, such as Fast-RCNN,

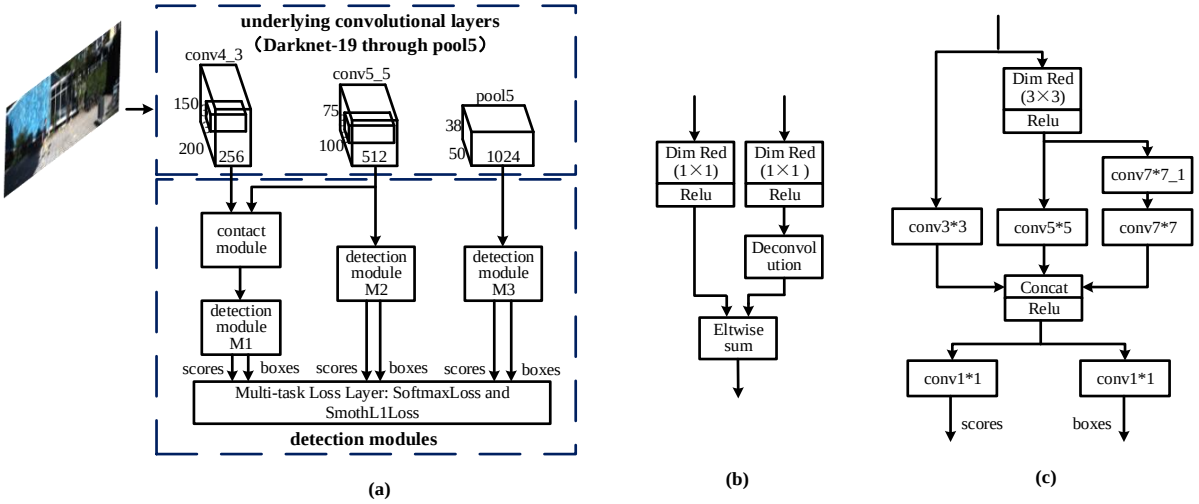


Figure 1 Architecture of Fast Single Stage Pedestrian Detector (FSSPD) **(a)** The FSSPD architecture: The structure is combined by underlying convolutional layers and detection modules. Note: the 'Dim Red' layer is a convolutional layer reducing the dim of a feather map. **(b)** Contact module architecture: The detail of a contact module which combines the feathers from conv4_3 and conv5_5. **(c)** Detection module architecture: The detail of a detection module which contains convolutional layers with three different kernel sizes 3*3, 5*5 and 7*7.

manually select these parameters from the width and height distribution of bounding boxes or other prior knowledge, while [22] automatically generated it with clustering algorithm for object detection in order to improve precision.

In this work, we will propose a single stage CNN based on Darknet19 instead of VGG16 to get higher speed and Darknet19 is modified via the use of a scale invariant network structure from SSH for a high accuracy. Especially for high-density pedestrian detection, we will adopt a dense anchor strategy to automatically select default anchor boxes with both big and small sizes in multi detection modules by using k-means clustering algorithm.

2. Fast Single Stage Pedestrian Detector

This section introduces our fast single stage pedestrian detector. To improve detection speed and accuracy, fast single stage framework designed in Sec.2.1 contains modified Darknet19 and multi detection modules. Since region proposing is significant for detection accuracy on this fast single stage framework, dense anchor strategy consisting of the default anchor boxes design strategy and unity anchor boxes strategy is presented in Sec.2.2. Moreover, to properly adjust the parameters of proposed fast single stage framework, training methods including multi-scale training strategy and OHEM are adopted in Sec.2.3.

2.1 Fast Single Stage Framework

Many single stage frameworks on the base of an anchor structure are widely used in object detection with their pros and cons. YOLOv2 has gotten an impressing result on Pascal VOC dataset but is inaccurate for small objects. While SSH is a state-of-the-art method for detecting small objects such as small face, but a VGG-16- based structure makes it working somewhat slowly. Our network will take the advantages of both YOLOv2 and SSH to meet the requirements of speed and accuracy in pedestrian detection.

General Architecture: Our framework shown in figure 1 can be divided to underlying convolutional layers and detector modules. As a basic feather extractor, the former based on the modified Darknet19 is pre-trained on ImageNet and then fine-tuned for the pedestrian detection task to speed up the training phase. The latter inspired by SSH detects objects from three different convolutional layers and can output detection results simultaneously.

Underlying convolutional layers: We remove the head of Darknet19 and reserve the layers from conv1 to pool5 to build our underlying network.

Detector modules: Ours detector modules M1~3 from SSH set anchor strides to 8, 16 and 32,respectively, by choosing conv4_3, conv5_5 and pool5 as detection layers for scale invariant, and each one can be divided into two synchronous working parts, region proposing and object classification. During the training phase, all parameters of detector modules are update at the same time. At the inference time, according to the outputs from different detector modules, Non Maximum Suppression (NMS) is used to form the final detections.

2.2 Dense Anchor Strategy

Due to the difficulty in high-density pedestrian detection, dense anchor strategy is proposed in this

paper, which contains two parts, the default anchor boxes design strategy and unity anchor boxes strategy.

Default anchor boxes design strategy: In the anchor-based CNNs, default anchor boxes are often manually selected for region proposing [13, 16, 21] and then the network can learn to adjust the positions and sizes of these boxes. While YOLOv2 uses the priors of the bounding boxes' width and height to get default anchor boxes. This strategy improves the algorithm accuracy and is adopted in our work.

To obtain the priors of bounding boxes, k-means as a classic and efficient clustering algorithm is used. Usually, the classic k-means with Euclidean distance may not get good IOU scores that we really want. So IOU distance [23] is a good substitution of Euclidean distance as

$$\mathrm{d}(c, b) = 1 - \mathrm{IOU}(c, b) \quad (1)$$

where *c* and *b* represent the sizes of clustering box and bounding box, respectively, and IOU(*c*, *b*) equals that the intersection area of boxes in sizes of *c* and *b* divided by their union area, given that these two boxes have same center.

Average IOU (Avg IOU) scores can be used to performance judgement on the two distances. Number choice of cluster center should take computation complexity and Avg IOU score into account.

Unity anchor boxes strategy: The sizes of default anchor boxes varies with the stride sizes of detector modules in SSH. Usually, a detector module with small anchor stride also has small default anchor boxes, and cannot deal with the special case of large objects at dense distribution such as pedestrian in a crowd. In this paper, unity anchor boxes strategy is proposed to let all the detector modules contain both big and small default anchor boxes.

2.3 Training

In the training phase, to update the network parameters, we use a stochastic gradient descent algorithm where there are momentum and weight decay tragedies to speed up the training and suppress overfitting, respectively. Since our network has three detection modules and each gets its own region proposing loss and classification loss, the multi-task loss of the whole network is

$$L(x, c, l, g) = \sum_k \frac{1}{N_k}(L_{conf}(x_k, c_k) + \alpha L_{loc}(x_k, l_k, g_k)) \quad (2)$$

where L_{conf} is the Softmax loss for object classification, and L_{loc} the SmoothL1 loss for region proposing indicating the difference between the detection boxes and the bounding boxes. N_k represents the detection box number of the *k*-th detection module, and x_k indicates whether the corresponding detection boxes matches the bounding boxes. c_k represents the probability that the detection boxes belongs to each category, l_k and g_k represent the position and size of the detection boxes and the bounding boxes, respectively.

To improve the performance for small targets, in the training phase we use multi-scale training strategy [18] to resize the input image by specifying the maximum length and width. Given an input image (w**h*) and the specified length h_m and width w_t, then the target size is

$$(w_f, h_f) = \begin{cases} (w, h) \times \frac{w_t}{w} & \frac{hw_t}{w} \le h_m \\ (w, h) \times \frac{h_m}{h} & else \end{cases} \quad (3)$$

This method can enlarge the input image of small target with a fixed aspect ratio. In addition, OHEM [23] is also used to improve the accuracy of network by removing some background detection boxes that have low classification losses.

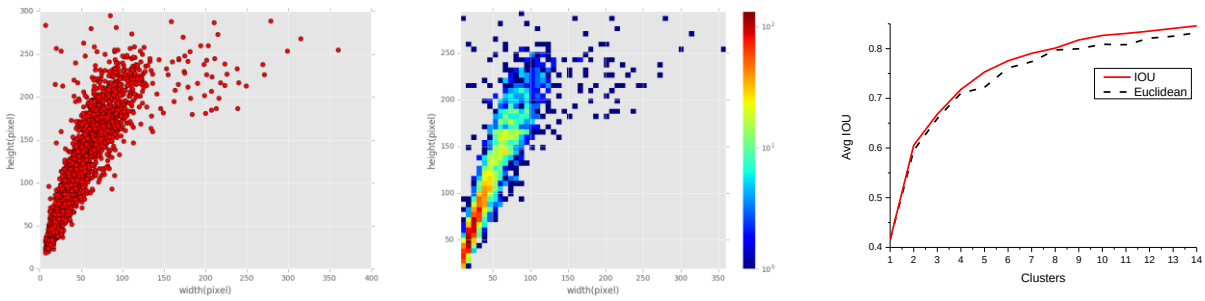
3. Experiments and Result Analysis

In this section, the proposed algorithm is evaluated and tested on the KITTI dataset. Section 3.1 describes the KITTI dataset. Section 3.2 includes experiments for clustering the bounding boxes, and FSSPD is evaluated on evaluation and test datasets, and compared with the original YOLOv2.

3.1 The KITTI Dataset and Its Metrics

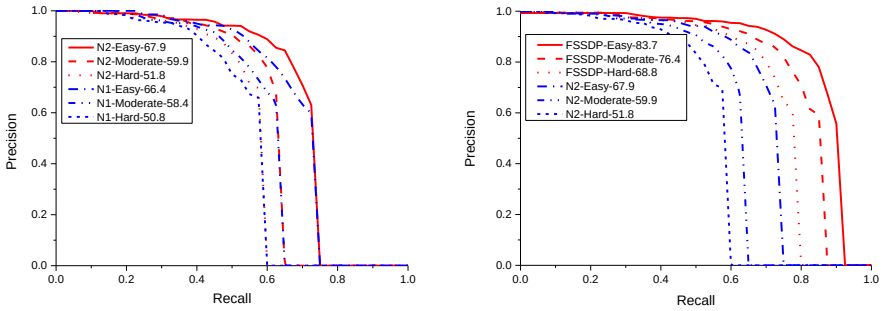
The KITTI dataset [24] dedicating to autonomous driving is collected by the driverless test vehicle during actual driving and contains a large number of small pedestrians that is difficult to detect, as shown in Figure 2. According to size, occlusion, and truncation, pedestrian detection is divided into three levels, i.e. easy, moderate, and hard, which is often used to test an algorithm performance on the different difficulties, as shown in Table 1.

Figure 2 Some pictures in the KITTI data set



(a) (b) (c)

Figure 3 The distribution of the train set bounding boxes and their clustering results. **(a)** Scatter plots for all bounding boxes of the train set. **(b)** Density plots generated from scatter plots, where the deeper the color, the greater the density. **(c)** The Avg IOU-Clusters curve when using different distance.



(a) (b)

Figure 4 Comparative experimental results on the evaluation set. **(a)** Precision-Recall curve of N2 and N1 **(b)** Precision-Recall curve of FSSDP and N2.

Table 1 KITTI dataset difficulty level classification standard

	Min. bounding box height	Max. occlusion level	Max. truncation
Easy	40px	Fully visible	15%
Moderate	25px	Partly occluded	30%
Hard	25px	Difficult to see	50%

In the KITTI dataset, Average Precision (AP) and Frame Per Second (FPS) are used as the evaluation metrics, besides, the overlap rate between detection boxes and bounding boxes greater than 50% is considered as a correct pedestrian detection.

3.2 Experiments

This section contains clustering and comparative experiments, both using GTX1080 and i7-5930k. As default anchor boxes should approximate to bounding boxes as possible, it's often manually designed from the width and height distribution of bounding boxes, while the dense anchor strategy proposed automatically finds default anchor boxes by k-means clustering algorithm. For generating the proper default anchor boxes with this two kinds of strategies, clustering experiments are designed to analyze the width and height distribution of bounding boxes in the train set, and compare IOU distance with Euclidean distance in k-means clustering. Based on clustering experiment, comparative experiments validates the effectiveness of the dense anchor and multi-detection-module strategy, and contain a comparison between FSSDP with YOLOv2.

(1) Clustering Experiments

The KITTI dataset contains 7481 annotated images and 7518 test images. During training, the annotated images are divided into a train set and an evaluation set, which are used to train the network and adjust hyper parameters in a ratio of 4:1, respectively.

For the train set, all the annotated images are clustered by k-means with IOU distance and Euclidean distance. The distribution of the train set bounding boxes and their clustering results are shown in Figure 3.

As shown in Figure 3(a) and 3(b), the bounding boxes widths distributes range from 0 to 300 and the main aspect ratio is about 2:1. Meanwhile, Figure 3(c) shows

that k-means algorithm with IOU distance has a higher Avg IOU score than with Euclidean distance, which indicates that IOU distance is more effective than Euclidean distance. And the Avg IOU score increases slowly with the increase of cluster center number, *k*. Taking the tradeoff of complexity and performance into consideration, k=5 is a good choice and now the clustering center result is {[15,42], [26,70], [43,108], [70,164], [124,217]}.

(2) Comparative Experiment

During training, the Darknet-19 network is pre-trained on ImageNet and the hyper parameters are set as follow: learning rate, momentum and weight decay are set to 0.0001, 0.9 and 0.005, respectively. And for all bounding boxes, anchor box with IoU<0.3 is considered as a negative sample, while with IoU>0.5 as a positive sample. Moreover, h_m and w_t are respectively set to 1200 and 1600. During inference, the anchors with the largest 1000 classification scores are selected as detection results, and the NMS threshold is set to 0.3.

Strategy effectiveness test: Three kinds of networks are evaluated on the evaluation set and their configures are shown in Table 2. N1 is a base network with only one detection module M3 and manually selected default anchor boxes. For validating the effectiveness of dense anchor strategy, N2 is designed with M3 and automatically selected default anchor box by dense anchor strategy. Moreover, in order to verify the multi-detection-module strategy, FSSDP contains multi detection modules consisting of M1, M2 and M3.

Table 2 Three kinds of networks configures

Network name	Detection module(s)	Dense anchor strategy
N1	M3	No
N2	M3	Yes
FSSPD	M1, M2, M3	Yes

According to the result of clustering experiment, the default anchor boxes for N1 is optimally set to {[22,44], [44,88], [88,176], [176,352], [352,704]}. For adapting the dense anchor strategy, that boxes for N2 and FSSDP is set on the basis of the results of k-means with IOU distance when k=5, that is, {[15,42], [26,70], [43,108], [70,164], [124,217]}.

Figure 4 shows the Precision-Recall curves of N1, N2 and FSSDP on the evaluation set. It can be seen that with the addition of dense anchor strategy and detection modules, the better performance is achieved for the samples at easy, moderate and hard levels, and the performance under hard level is still not ideal. To meet the requirement of rapidity, all three networks adopt single stage detecting structure, which is difficult to detect small and occluded pedestrians at hard level.

Figure 4(a) shows the performance comparison between N1 and N2. In three difficulty levels, N2 improved the average precision by 1.5%, 1.5% and 1% than N1, respectively. As shown in Table 2, N2 uses the proposed dense anchor strategy, since default anchors are chosen automatically by k-means clustering algorithm instead of manual selection and contain both big and small sizes in all detector modules, it can be seen that dense anchor strategy is effective for the improvement of average precision.

From Figure 4(b), it can be seen that FSSDP outperforms N2 by 15.8%, 16.5% and 17% in easy, moderate and hard level, respectively. As shown in Table 2, FSSDP contains M1, M2 and M3 with anchor strides 8, 16 and 32, respectively, the multi-detection-module strategy is good for scale invariant detection and improves average precision.

Compared Figure 4(a) to 4(b), the increase of AP in Figure 4(a) is greatly smaller than that in Figure 4(b). It is shown that dense anchor strategy is less effective than multi-detection-module strategy. Moreover, the improvement for hard level is larger than that for easy and moderate levels, indicating that multi-detection-module strategy is more suitable for detecting small and occluded pedestrians.

Comparison with YOLOv2: The proposed network is compared with the YOLOv2 network on the test set. As shown in Figure 5 and Table 3, FSSDP improves average precision by 39.86%, 30.96%, and 27.25% than the YOLOv2 algorithm in the easy, moderate and hard levels, respectively, while achieves pedestrian detection at the speed of 14.3fps in near real time. This indicates that FSSDP is suitable for pedestrian detection at a good tradeoff between precision and speed.

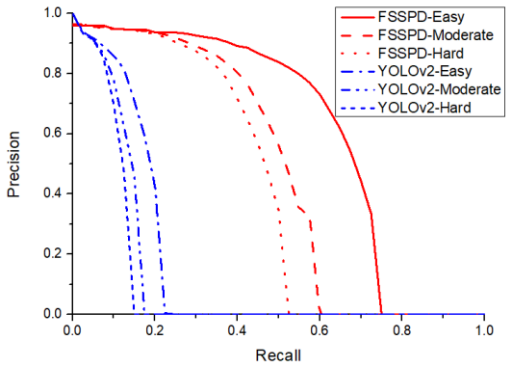


Figure 5 Precision-Recall curve between YOLOv2 and FSSPD on the test set.

Table 3 Comparative experimental results between YOLOv2 and FSSPD on the test set.

Network name	Easy (AP)	Moderate (AP)	Hard (AP)	Inference time(s)
YOLOv2	20.80	15.43	16.19	0.02
FSSPD	60.66	46.39	43.44	0.07

4. Conclusions

Aiming at small and high-density pedestrian detection for autonomous driving, FSSDP network is proposed by using the modified Darknet-19 structure with dense anchor and multi-detection-module strategies. To improve the speed of pedestrian detection, Darknet-19 structure from fast object detection CNN YOLOv2 is used. For high average precision, dense anchor and multi-detection-module strategy are proposed. The clustering and comparative experiments show that the proposed network structure and strategies are effective for small and high-density pedestrian detection, and the multi-detection-module strategy has a significant improvement for small pedestrian. Finally, the comparison between FSSDP with YOLOv2 indicates that FSSDP is more suitable for pedestrian detection taking precision and speed into account.

Acknowledgement

This research was supported by Primary Research& Development Plan of Jiangsu Province (No. BE2016155), the National Natural Science Foundation of China (No. 51477125) and the Hubei Science Fund for Distinguished Young Scholars (No.2017CFA049).

Reference

[1] Geronimo D, Lopez A M, Sappa A D, et al. Survey of pedestrian detection for advanced driver assistance systems. IEEE transactions on pattern analysis and machine intelligence, 2010;32(7):1239-1258.
[2] Dollár P, Appel R, Belongie Set al. Fast Feature Pyramids for Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014;36(8):1532-1545.
[3]Ohn-Bar E, Trivedi M M, To Boost Or Not to Boost? On the Limits of Boosted Trees for Object Detection, Pattern Recognition (ICPR). 2016:3350-3355.
[4] Levi D, Silberstein S, Bar-Hillel A, Fast Multiple-Part Based Object Detection Using Kd-Ferns, Computer Vision and Pattern Recognition (CVPR). 2013:947-954.
[5] Zhang S, Benenson R, Schiele B. Filtered Channel Features for Pedestrian Detection. Proceedings of the IEEE Computer Vision and Pattern Recognition. 2015:4.
[6] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2005:886-893.
[7] Felzenszwalb P F, Girshick R B, Mcallester Det al. Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010;32 (9):1627-1645.
[8] Du X, El-Khamy M, Lee Jet al. Fused Dnn: A Deep Neural Network Fusion Approach to Fast and Robust Pedestrian Detection. Applications of Computer Vision, 2017 IEEE Winter Conference on: IEEE. 2017:953-961.
[9] Cai Z, Fan Q, Feris R Set al. A Unified Multi-Scale Deep Convolutional Neural Network for Fast Object Detection. European Conference on Computer Vision: Springer. 2016:354-370.
[10] Brazil G, Yin X, Liu X, Illuminating Pedestrians Via Simultaneous Detection & Segmentation, arXiv preprint arXiv:1706.08564, 2017, .
[11] Tian Y, Luo P, Wang Xet al., Pedestrian Detection Aided by Deep Learning Semantic Tasks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015:5079-5087.
[12] Zhang L, Lin L, Liang Xet al. Is Faster R-Cnn Doing Well for Pedestrian Detection? European Conference on Computer Vision: Springer. 2016:443-457.
[13] Ren S, He K, Girshick Ret al. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. Advances in neural information processing systems. 2015:91-99.
[14] Girshick R. Fast R-Cnn. arXiv preprint arXiv:1504.08083, 2015.
[15] Redmon J, Divvala S, Girshick Ret al. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:779-788.
16] Liu W, Anguelov D, Erhan Det al. Ssd: Single Shot Multibox Detector. European conference on computer vision: Springer. 2016:21-37.
[17] Redmon J, Farhadi A. Yolo9000: Better, Faster, Stronger, arXiv preprint. 2017.
[18] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556, 2014.
[19] He K, Zhang X, Ren Set al. Deep Residual Learning for Image Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:770-778.
[20] Szegedy C, Ioffe S, Vanhoucke Vet al. Inception-V4, Inception-Resnet and the Impact of Residual Connections On Learning. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017:4278-4284.
[21] Najibi M, Samangouei P, Chellappa Ret al. Ssh: Single Stage Headless Face Detector. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:4875-4884.
[22] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
[23] Shrivastava A, Gupta A, Girshick R. Training Region-Based Object Detectors with Online Hard Example Mining. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:761-769.
[24] Geiger A, Lenz P, Urtasun R. Are we Ready for Autonomous Driving? The Kitti Vision Benchmark Suite. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2012: 3354-3361.