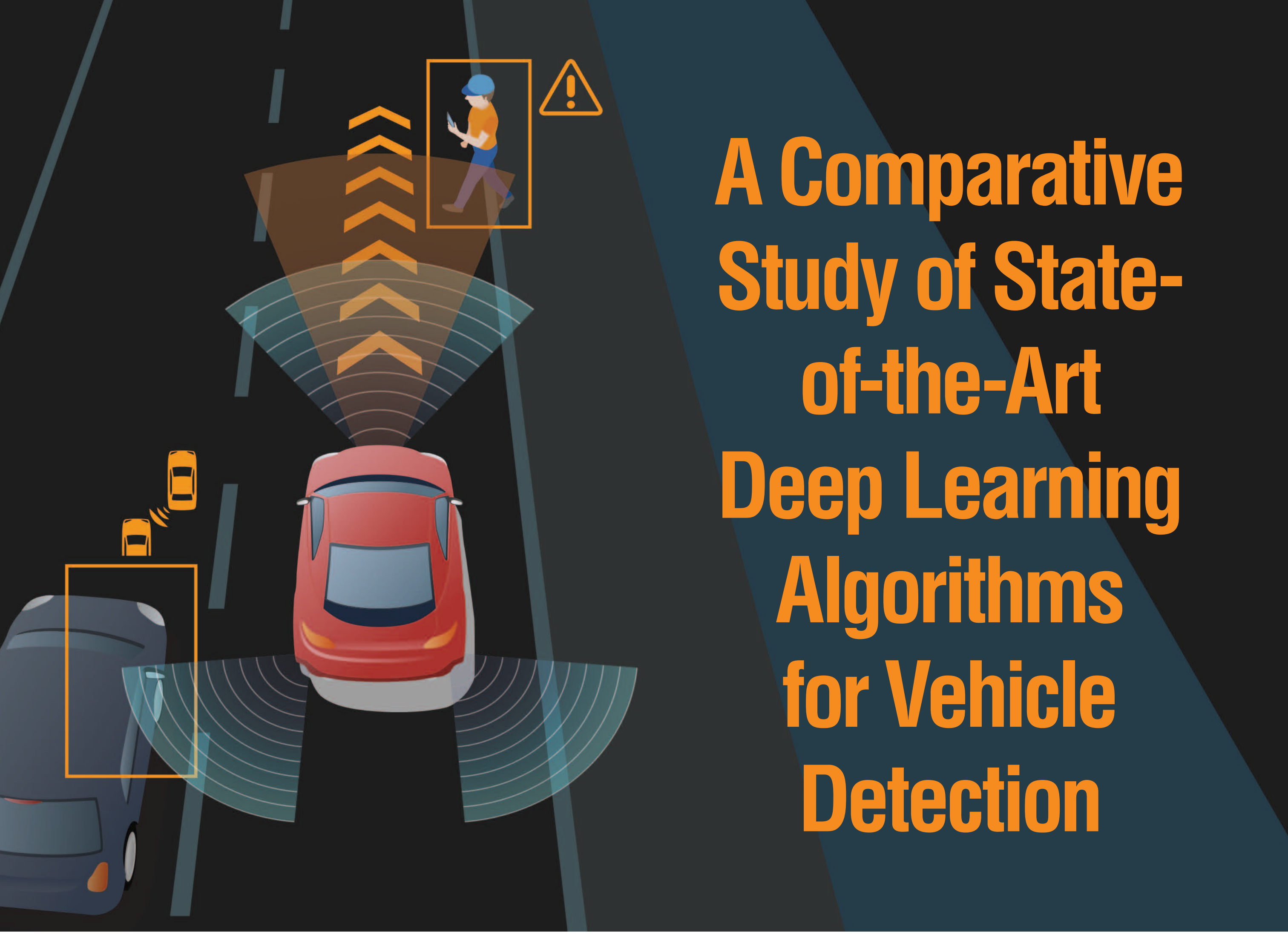


A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection

©ISTOCKPHOTO.COM/CHOMBOSAN

Hai Wang and Yijie Yu,
is with School of Automotive and Traffic Engineering of Jiangsu University,
Zhenjiang, 212013, China.
E-mail: wanghai1019@163.com, 13588280094@163.com

Yingfeng Cai, Xiaobo Chen, Long Chen, and Qingchao Liu
is with Automotive Engineering Research Institute of Jiangsu University,
Zhenjiang, 212013, China. E-mail: caicaixiao0304@126.com, xbchen82@gmail.com, chenlong@ujs.edu.cn, uiodo@hotmail.com

Digital Object Identifier 10.1109/MITS.2019.2903518
Date of publication: 18 March 2019

Abstract—In recent years, the deep learning object detection algorithms using 2D images have become the powerful tools for road object detection in autonomous driving. In fact, the deep learning methods for road vehicle detection have achieved the remarkable results. Although there have been a large number of studies that thoroughly explored various types of deep learning methods for vehicle detection, there are a few studies that compare and evaluate the detection time and detection accuracy of the mainstream deep learning object detection algorithms for vehicle detection. Here, this article compares five mainstream deep learning object detection algorithms in vehicle detection, namely the faster R-CNN, R-FCN, SSD, RetinaNet, and YOLOv3 on the KITTI data and analyze the obtained results. The detection time and detection accuracy of the five object-detection algorithms on the KITTI test set are compared and analyzed; the PR curve and AP value are used to evaluate the detection accuracy.

1939-1390/19©2019IEEE

I. Introduction

In the past ten years, the automobile industry has developed very rapidly, the vehicle navigation systems have become more and more common, and the degree of intelligence of vehicles has been continuously improved. From the L0-class vehicles with no automation, vehicles have developed to the partially-automated L2 level. However, there is still a long way to go to realize the real autonomous driving vehicles, but the overall demand for smart vehicles has become more and more obvious. Taking the initial level of the automatic driving level L3 as an example, an intelligent vehicle must at least not require a driver to monitor the current road condition in real time, and only need to take over a vehicle when prompted by the system. To make the navigation system complete, a self-vehicle can be positioned, and driving can be established. Accordingly, a driver becomes a passenger, and a passenger does not need to monitor the current road conditions in real time. Replacing the driver with all types of real-time road conditions monitoring systems, in the future, a potential interference to the driver due to the navigation console and other related infotainment systems can be greatly avoided, and the accidents caused by a driver distraction can be significantly reduced [1].

> In recent years, in the computer vision field, the real-time performance of deep learning vehicle detection algorithms has been comparable to that of the traditional manual feature-based vehicle detection algorithms due to a continuous increase in data volume and a rapid advancement of hardware. Moreover, the deep learning methods far exceed the traditional algorithms regarding the detection accuracy.

In a driver-assisted vehicle, all kinds of sensors are particularly important because a vehicle obtains the correct information from the sensors. Currently, sensors used in the smart vehicles mainly include the lidar, radar, monocular camera, and binocular camera. These four sensors have their advantages and disadvantages [2]. For instance, the drawback of radar is that the detection distance is relatively close. Besides, when a vehicle is driven at high speed, the radar detects objects at less than 100 meters. Thus, it is difficult for radar to analyze and predict the forward obstacles. On the other hand, a lidar does not work well in the extreme weather conditions, and it is costly. Further, both monocular and binocular cameras have drawbacks. In the case of the night or other poor light, the detection result is even worse. At present, the main obstacle to the launch of intelligent vehicles is a high production cost; namely, the cost of the monocular and binocular camera is relatively low, but a laser radar is much more expensive. A vision-based intelligent driving scheme is cheaper than a lidar-based scheme so that it can be put on the market earlier. Consequently, the study on the vision-based vehicle detection solutions is particularly important [3]–[4].

In recent years, in the computer vision field, the real-time performance of deep learning vehicle detection algorithms has been comparable to that of the traditional manual feature-based vehicle detection algorithms due to a continuous increase in data volume and a rapid advancement of hardware. Moreover, the deep learning methods far exceed the traditional algorithms regarding the detection accuracy.

The contributions of this paper are as follows. The five types of currently mainstream deep learning object detection algorithms (faster R-CNN [5], R-FCN [6], SSD [7], RetinaNet [8], and YOLOv3 [9]) are compared on the KITTI benchmark [10]. The parameters of the five algorithms are chosen such that to ensure the credibility of the experimental results, and the detection algorithms are compared regarding the detection time, recall, and precision. The obtained test results are analyzed and summarized, which have a certain guiding significance for research in the vehicle detection field.

The remainder of this paper is organized as follows. Section II presents the works related to the road vehicle detection. Section III explains the background and working principle of the five mainstream deep learning object detection algorithms. Section IV provides the analysis of the experimental results obtained by the vehicle detection algorithms and presents some predictions and analysis of future vehicle detection algorithms. Section V provides the conclusions.

II. Related Works

A. Traditional Vehicle Detection Algorithms

Using the computer vision to detect other vehicles on the road accurately is a challenging task and has been a hot research topic in the past two decades [11]. The roads the vehicles travel on are dynamic with constantly changing background and lighting. At the same time, all vehicles on the road are usually moving, so the size and position of a vehicle in an image plane obtained by a camera are various. The shape, size, and color of a vehicle found in typical driving scenarios are highly variable. Vehicle detection and tracking have been widely explored in the literature for more than a decade. In earlier studies, various artificially extracted features were used for vehicle detection. The three most commonly used features were Haar [12], HOG [13], and LBP [14]. The classifiers matched by these three features which mainly include the support vector machine (SVM) [15], Adaboost [16], and the Haar feature set combined with the Adaboost, were widely used in the computer vision methods initially intended for face detection [17] and various follow-up applications. The classification framework was applied to the vehicle detection and found to have a good performance (the number of positive samples correctly detected is large). The HOG feature combined with the SVM classifier has been widely used in image recognition [18], and it has achieved great success in pedestrian detection. In addition to the above-mentioned several features and classifiers with wide applicability [19]–[20] to the vehicle detection tasks, the statistical models based on the vertical and horizontal edge features were proposed for vehicle detection [21] and tracking of vehicles at night by positioning the tail lights.

B. Vehicle Detection Algorithms Based On Deep Learning

In 2012, a CNN network named the AlexNet [22] won the ImageNet image recognition competition; its top-5 error rate was only 15.3%, which was 26.2% smaller than the second-placed SVM method. Recently, the deep learning methods have attracted the attention of many researchers, and a large number of deep learning object detection algorithms have emerged. Compared with the traditional methods, in the deep learning object detection algorithms, manual extraction of features requires experts with years-long expertise in the related domain. The deep learning methods require a large amount of data and automatically learn the characteristics that can reflect the difference in data, making it more representative. At the same time, in visual recognition, the process of a CNN layer feature extraction is similar to the human visual mechanism, which represents the process from the edge to a part to the whole [23]. In recent years, with the continuous expansion in data volume and constant update of devices' hardware, the deep learning target detection algorithms have obtained competitive real-time performance compared with the traditional methods, and have begun to gain the recognition from the industry worldwide. In the academic field, due to the different emphasis on the real-time performance and accuracy, the deep learning object detection algorithms have gradually developed in two directions, two-stage object detection focused on the detection accuracy, and one-stage object detection focused on the detection speed. Table I shows the innovations and shortcomings of many object detection algorithms in recent years including the two-stage and one-stage algorithms.

Table I. Innovations and defects in related work.

Algorithms		Innovations	Defects
Two-stage	R-CNN[24]	Abstracts the object detection into two processes The first one is using the selective search algorithm [25]; the second one is to apply the classification network on these proposed regions to get the category of objects within a zone.	The Selective Search algorithm runs too slowly
	Fast-RCNN[26]	Optimizes slow detection of an R-CNN algorithm; it proposes to transfer the basic network to the R-CNN subnet after processing the whole image	Still unable to achieve real-time detection (detecting more than 5 frames per second)
	Faster-RCNN	Proposes to replace the selective search algorithm with the RPN network so that the detection task can be completed end-to-end by a neural network	Detection speed is not as good as one-stage algorithm
	R-FCN	Modified the structure of the faster R-CNN to solve the contradiction between translation invariance and position sensitivity partially	Detection speed is not as good as one-stage algorithm
One-stage	YOLO[27]	Transform the inspection task into a uniform, end-to-end regression problem, and obtain the position and classification simultaneously by only one process. Compared to the two-stage object detection algorithms, the YOLO has a faster detection speed	Poor detection accuracy
	SSD	Through the multi-scale feature map extraction, the algorithm achieves better detection effect on small objects; the SSD generates more anchor points to make the object positioning more accurate	Poor accuracy for small-scale target detection
	YOLOv2[28]	A new feature extraction network (Darknet-19), an adaptive anchor box, and multi-scale training	Poor accuracy for small-scale target detection
	YOLOv3	Introducing a multi-scale network structure similar to the FPN [29], and by referring to the Resnet [30], the detection accuracy for small objects was further improved	Detection accuracy is still not as good as two-stage algorithm
	RetinaNet	Proposed the RetinaNet with a focal loss solve this problem that the loss of a large number of difficult cases can be covered by the loss of simple samples	Detection speed is not as good as YOLO and SSD algorithm

III. Principles of Mainstream Vehicle Detection Algorithms

In recent years, a large number of excellent-performance methods have emerged in the deep learning based object detection field. These methods take into account the specificity of road vehicles, such as the facts that driving roads are dynamic environments, the lighting and background in the environments change a lot, the dimensions and positions of vehicles in the image are diverse, and the shape, size and color of the vehicle are different. The automatic driving has high requirements for real-time performance of the object detection algorithms. This article has selected five mainstream object detection algorithms from many object detection algorithms to conduct comparative research on the detection algorithms with outstanding accuracy and real-time performance.

This article has selected five mainstream object detection algorithms from many object detection algorithms to conduct comparative research on the detection algorithms with outstanding accuracy and real-time performance.

A. Two-Stage Vehicle Detection Algorithms

The two-stage object detection models include the two-stage processing of an image, and they are also known as the region-based object detection methods. The main idea is first to generate a series of sparse candidate frames, and then, to classify and regress the candidate frames. The advantage of the two-stage object detection methods is high accuracy. In this work, the faster R-CNN and the R-FCN are used as the representative algorithms of this type of object detection models.

1) Faster R-CNN

Before the faster R-CNN was proposed, the most advanced vehicle detection models used the selective search to estimate target location. The networks such as the SPPnet [31] and fast R-CNN have reduced the operation time of a detection network, but the calculation is still time-consuming. Compared with the R-CNN and fast R-CNN, in the faster R-CNN, four basic steps of object detection, namely the candidate region generation, feature extraction, classification, and location refinement, are unified into a deep network framework because the candidate region selection algorithm (the Selective Search) takes more time during the detection. The faster R-CNN proposes to use a CNN, the RPNs (Region Proposal Networks), to select the candidate regions. The RPN network first generates region proposals, and then classifies them, which is called the two-stage processing, as shown in Fig. 1. The author observed that the feature map of the convolutional layer of the region detector in the fast R-CNN could be used to generate the candidate regions for the RPNs. On the basis of feature mapping, several convolutional layers are added backward to form a regional recommendation network, which denotes a fully convolutional network (FCN) [32].

The basic idea of the RPN network is as follows. In the extracted feature map, the feature vector is obtained through a sliding window and then output to two layers, bounding box regression layer and bounding box classification layer.

There is a sliding window on the feature map, which traverses every point on the feature map and configures k anchor boxes at each point, as shown in Fig. 2. These k

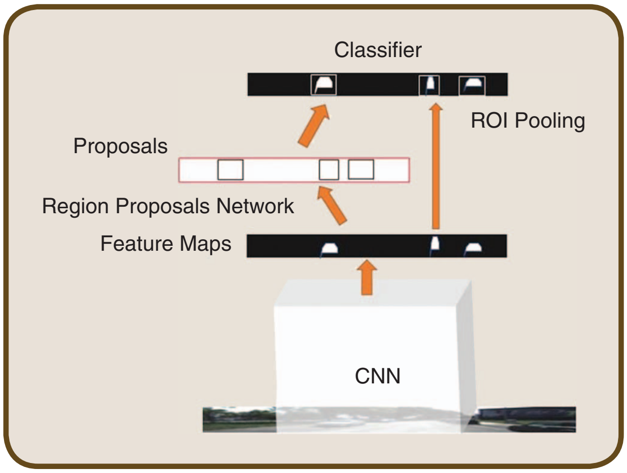


FIG 1 Framework of the faster R-CNN [5].

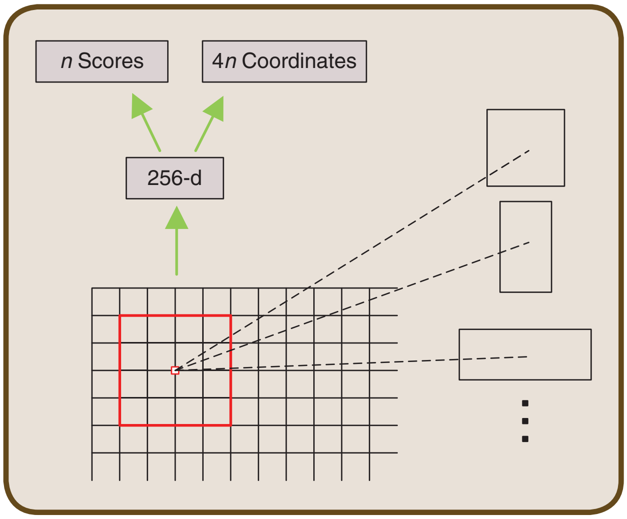


FIG 2 Framework of the RPN [5].

anchor boxes are used to extract features of the feature map, but the effect is not very good, so a classifier and a box regression should be used. There are two parallel loss functions, *softmax* and *smoothL1*, that classify and regress each region of interest (RoI) respectively. In this way, the model can get a real category and more precise coordinates, length, and width of each RoI.

Finally, using the VGG16 [33] as a basic feature extraction network, the performance of faster R-CNN was improved from 39.3% to 42.1% compared with the fast R-CNN on the COCO test set [34].

2) R-FCN

The two-stage object detection network can often be divided into two parts. The first part denotes the basic classification network such as the VGG, GoogleNet [35], ResNet, etc. The calculation of these networks is shared by all RoIs, and only one forward calculation is required in the image test. In the sub-network of the second part, the calculation it is not shared by all RoIs because this part aims to classify and return each RoI, so the calculation cannot be shared since the positioning accuracy of the two-stage object detection network could be affected. Specifically, the first network part has position invariance. For instance, all the convolutional layers of the ResNet are placed in the first part to extract features, while the second part has only the fully connected layer. However, for vehicle detection, the information on a vehicle position in an image is also very important, which is one of the reasons why the accuracy cannot be improved further.

With the aim to solve this problem, the R-FCN was proposed. The R-FCN has a modified structure of the faster R-CNN and partially solves the contradiction between position invariance and position sensitivity by maximizing the sharing of convolution parameters. The training and testing efficiency has been dramatically improved under the same precision. The framework of the R-FCN is presented in Fig. 3.

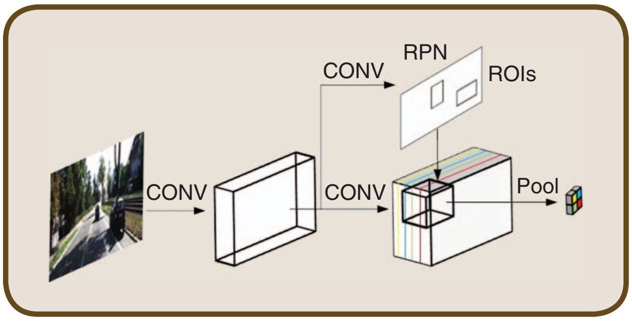


FIG 3 Framework of the R-FCN [6].

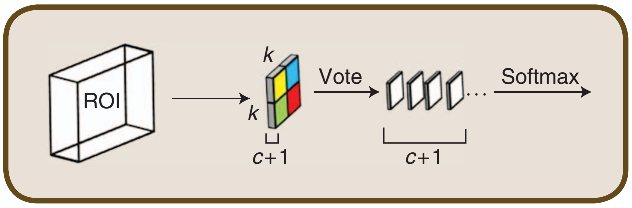


FIG 4 Framework of the ResNet [6].

The main idea of the R-FCN algorithm is to establish the “position-sensitive score maps”. A generated RoI must have $(k \times k)$ sub-regions containing the corresponding parts of the object to determine whether the RoI belongs to the object and if there are many parts of the object appearing in the corresponding sub-area, then, the RoI is judged to be the background category. In order to achieve this goal, the R-FCN is connected to a convolutional layer at the end of the shared convolutional layer, which denotes the “position-sensitive score maps”. First, the height and width of this convolutional layer are the same as those of the shared convolutional layer, but its channel number is $k^2 \times (C+1)$, where C represents the number of object category plus one background category, and each category has $(k \times k)$ score maps. Assuming the category denotes a car, then, there are $(k \times k)$ score maps, and each score map denotes “the position of a part of the car in the original image”, and the score map has a “high response value” at the location containing “a part of the vehicle corresponding to the score map”. In order to find the corresponding value of each sub-region in each category of the score maps, this article put forward the idea of “position-sensitive RoI pooling”, and the RoI region extracted by the RPN. It contains four attributes, namely x and y coordinates, length and width; namely, different RoI areas can correspond to different positions of the score map, and one RoI is divided into $(k \times k)$ areas, each of which is corresponding to the one area of the score map.

Accordingly, one RoI gets $(k^2 \times (C+1))$ values, as shown in Fig. 4. For each category, $(k \times k)$ values of the category indicate that the RoI belongs to the response value of the category, adding $(k \times k)$ numbers to obtain the score of the category, so there are a total of $(C+1)$ scores. Then, using the simple *softmax* function for $(C+1)$ scores, which can get the probability of belonging to each category. The bounding box regression also uses a similar idea to get the four predicted values for regression. According to the idea of “position-sensitive score map” and “position-sensitive ROI pooling”, a score map, the “position-sensitive score map”, for regression is designed in this work. In the last layer of the shared convolutional layer of the ResNet, a score map parallel to the “position-sensitive score map” is used for the bounding box regression, named the “regression score map” with the dimension of $(4 \times k \times k)$. After the position-sensitive RoI pooling operation, each RoI also gets four numbers as the offset of the corresponding RoI’s coordinates, length, and width.

Finally, the comparison of the R-FCN and the faster R-CNN on the VOC [36] and COCO datasets showed that the

R-FCN had better detection accuracy and the detection speed was faster than that of the faster R-CNN.

B. One-Stage Vehicle Detection Models

In the one-stage object detection models, there is no intermediate region detection process and prediction results are obtained directly from an image. The main idea is to sample an image at different positions uniformly. Different scales and aspect ratios can be used for sampling. After sampling, a CNN is used to extract features and direct classification and regression; the whole process only needs one step, so its advantage is a fast operation. Here, the SSD, YOLOv3, and RetinaNet are used as representative algorithms of one-stage object detection models.

1) SSD

Compared to the earliest single-stage target detection model, such as the YOLO, the SSD uses a convolutional layer to detect an object directly, rather than to detect it after the fully connected layer as in the YOLO. In addition, there are two important innovations: (1) the SSD extracts feature maps of different scales for detection, where the large-scale feature maps (more advanced feature maps) can be used to detect small objects, and the small-scale feature maps (later feature maps) are used to detect large objects; (2) the SSD such as the faster R-CNN adopts an anchor boxes of different scales and aspect ratios, as shown in Fig. 5. The disadvantage of the YOLO algorithm is that it is difficult to detect small objects, and the positioning is not accurate. The SSD can overcome these shortcomings to a certain extent.

The SSD uses the VGG16 as a basic feature extraction network, and then adds more convolution layers on the VGG16 to obtain more feature maps for detection. Specifically, given an input image and a series of truth labels, the SSD passes the image through a series of convolutional layers, producing a series of feature maps of different sizes, and then, for each of these feature maps, the position uses a 3 × 3 convolution filter to evaluate some of the default bounding boxes. Each of these bounding boxes simultaneously performs the prediction of the bounding box's offset and the probability of classification. During the training, these prediction bounding boxes based on the IOU coefficients are used to match the correct bounding box. The best-predicted bounding box will be labeled as a positive sample. The framework of the SSD is given in Fig. 6.

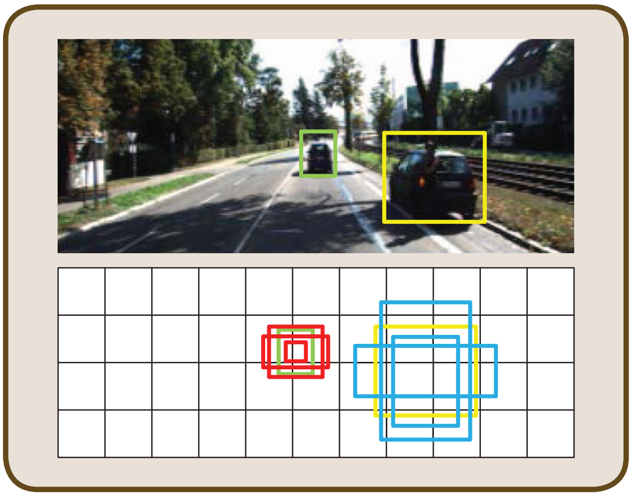


FIG 5 The anchor generation of the SSD [7].

In this work, the SSD is compared with its previous target detection algorithm using the VOC2007 test set as a benchmark, and it is found that SSD has the same accuracy as the faster R-CNN and the same detection speed as the YOLO.

2) RetinaNet

Due to the lack of the regional proposal operations such as those in two-stage object detection models, the detection algorithm of one-stage object detection models often has an order of magnitude higher/larger than that of two-stage object detection models, and these regions have extremely uneven categories. The loss of a large number of difficult cases will be covered by the loss of simple samples. This is one of the main causes why the accuracy of the one-stage object detection models is not as good as that of the two-stage object detection models. To solve this problem, He et al. proposed a focal loss to replace the standard binary cross entropy of the original classification, which is equivalent to adding a respective weight to each sample. This weight is related to the probability that the network predicts that

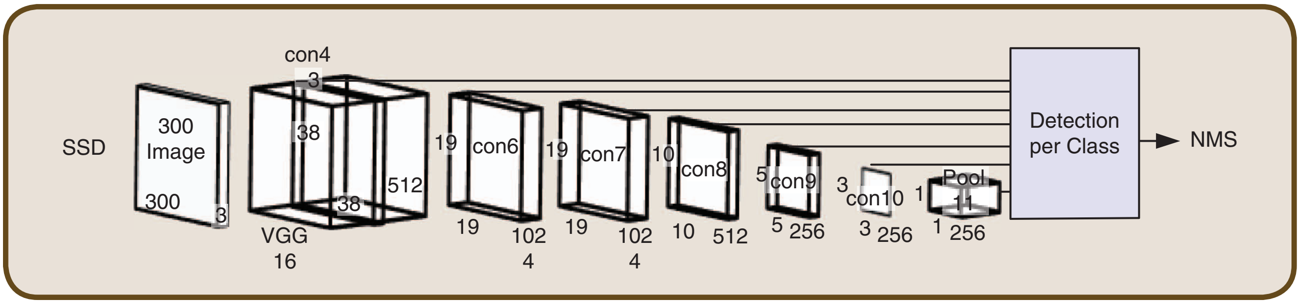


FIG 6 Framework of the SSD [7].

the sample belongs to the real classification. If the probability that the network predicts that the sample belongs to the real classification is large, then that sample denotes a simple sample for the network; otherwise, if the probability that the network predicts that the sample belongs to a real sample is small, then that sample denotes a difficult sample for the network. In order to create an effective classification model, the weights of most simple samples should be reduced, and the weights of difficult samples should be increased relatively. Usage of a focal loss function can make the loss drop faster and reach a smaller value, improving the classification performance of the model, so that a one-stage model can achieve or even exceed a two-stage model in terms of the accuracy. To demonstrate the effect of a focal loss, here, a new one-stage object detection model, the RetinaNet, which uses the multi-scale idea of the Resnet network combined with the FPN as a basic feature extraction network, is presented, and it connects the two sub-networks of the classification and bounding box regression frame to each multi-scale output, as shown in Fig. 7.

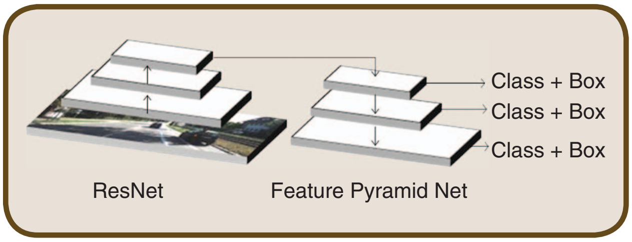


FIG 7 Framework of the RetinaNet [8].

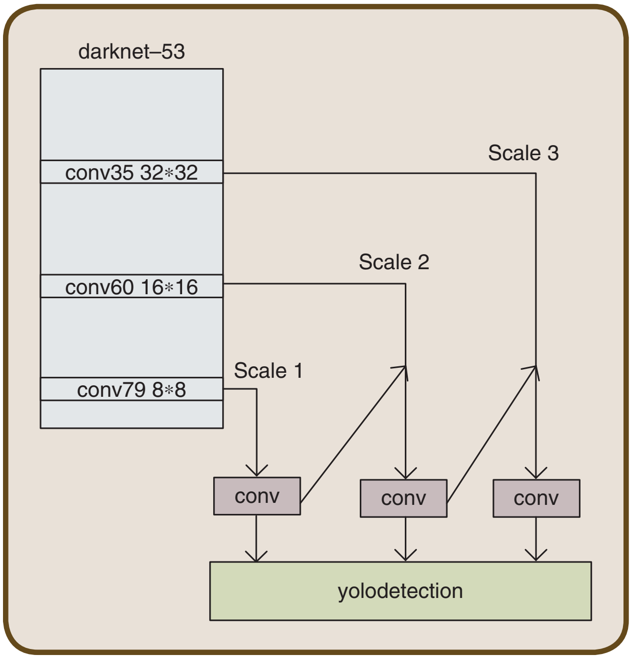


FIG 8 Framework of the YOLOv3 [9].

Based on the above improvements, the focal loss was compared with the OHEM [37], which is also used to solve the problem of an insufficient weight of difficult samples (the OHEM is equivalent to training a hard example). The result proved that the mAP value of a model using the focal loss greatly exceeded that of the OHEM method. This article also tested the RetinaNet model on the COCO dataset and compared the results with those of several mainstream object detection algorithms. The RetinaNet achieved the mAP value of 39.1 on the COCO dataset. In the case of a high detection speed, the detection accuracy of the RetinaNet was comparable to that of the two-stage object detection model.

3) YOLOv3

The YOLOv3 has been the most balanced object detection network regarding the real-time performance and accuracy. Through the fusion of various advanced methods, the shortcomings of the previous two generations of the YOLO object detection algorithms (low detection accuracy, unsatisfactory detection of small objects, etc.) are all overcome.

Firstly, the YOLOv3 draws on the idea of the Resnet, and it employs the Darknet-19 of the YOLOv2 and creates a new feature extraction network called the Darknet-53. The classification accuracy of the Darknet-53 is close to that of the ResNet-101 or ResNet-152, but the Darknet-53 is much faster.

At the same time, in view of a poor detection of small-scale objects of both YOLO and YOLOv2, the YOLOv3 draws on the idea of the FPN, as shown in Fig. 8. The anchor generation method of the YOLOv3 inherits the YOLOv2 using the k-means clustering method [38]. Then, nine cluster centers are obtained and equally divided into three scales according to the size. After that, by adding the convolutional layers after the basic network and outputting the prediction frame information this article obtains the first scale to target the large object by sampling from the convolutional layer of the penultimate layer in the scale 1; then, the last 16×16 size feature map is added, and the predicted bounding boxes information is output again after multiple convolutions. Compared with the scale 1 which is doubled, the accuracy of detecting the medium object can be improved, and the upsampling method for detection of small objects is similar to the detection of the medium objects. After upsampling, using the last 32×32 size feature map, the YOLOv3 can significantly improve the detection accuracy of small-scale objects. The medium objects are sampled by the convolutional layer of the penultimate layer in scale 1 and then added to the last 16×16 size feature map, and the predicted frame information is output again after multiple convolutions. This method improves the detection accuracy of medium objects, and the detection accuracy of small objects is similar to that of the medium objects. After upsampling, it is combined with the last 32×32 size feature map. In this way, the YOLOv3 can significantly improve the detection accuracy of small-scale objects.

In addition, another significant improvement of the YOLOv3 is the choice of classifiers. The YOLOv3 no longer uses the softmax function to classify each box because softmax assigns each box a category, and there may be overlapping category labels for the object, so the data set may not perform well. Therefore, in the YOLOv3 softmax is replaced by the independent multiple logical classifiers, and the classification loss is calculated using the binary cross entropy loss.

Due to the above-mentioned improvements, the YOLOv3 can achieve the mAP value of 57.9 on the COCO dataset, which is similar to that of the RetinaNet, but the YOLOv3 is nearly four times faster than the RetinaNet.

IV. Experimental Evaluation

The KITTI benchmark was used for evaluation of five object detection models. This article mainly evaluated the average accuracy of the models for road vehicles, the test time of a single picture, and the generalization ability of the actual road vehicle images, and these indicators were used to measure whether the object detection algorithm can be used for vehicle detection.

A. Data Description

The KITTI dataset selected for the experiment contained the real-world image data collected from different scenes such as urban roads, rural roads, and highways, as shown in Fig. 9, with up to 15 vehicles and 30 pedestrians per image, and these goals also had varying degrees of occlusion and truncation. The two-dimensional target detection part of the data set was selected as the training and test sample. The training set contained 7481 images and the corresponding label comments, and the test set contained 7518 images, which were divided into three difficulty-level groups: easy, moderate, and hard. The original data set contained eight types of objects, namely Car, Van, Truck, Pedestrian, Person, Cyclist, Tram, and Misc. In the experiment, the labels in the comments were reorganized as needed, and the irrelevant samples were eliminated. Besides, Car, Van, and Truck were retained as Cars.

B. Data Augmentation

In order to increase the amount of data and improve the generalization ability of the model without changing the image category, a series of augmentation operations were performed on the data set in the experiment, as shown in Fig. 10, including the image scaling, displacement, cropping, and noise adding, considering the attitude of a normal vehicle; this article did not use picture rotation and inversion.

C. Model Super-Parameter Selection

In the model training, the whole data set can be used as an input of the neural network, so that the neural network can use all the samples to calculate the gradient of an iteration; also, a few samples can be used as an input of the network to complete the local iteration, which

FIG 9 A Snapshot of the KITTI benchmark.

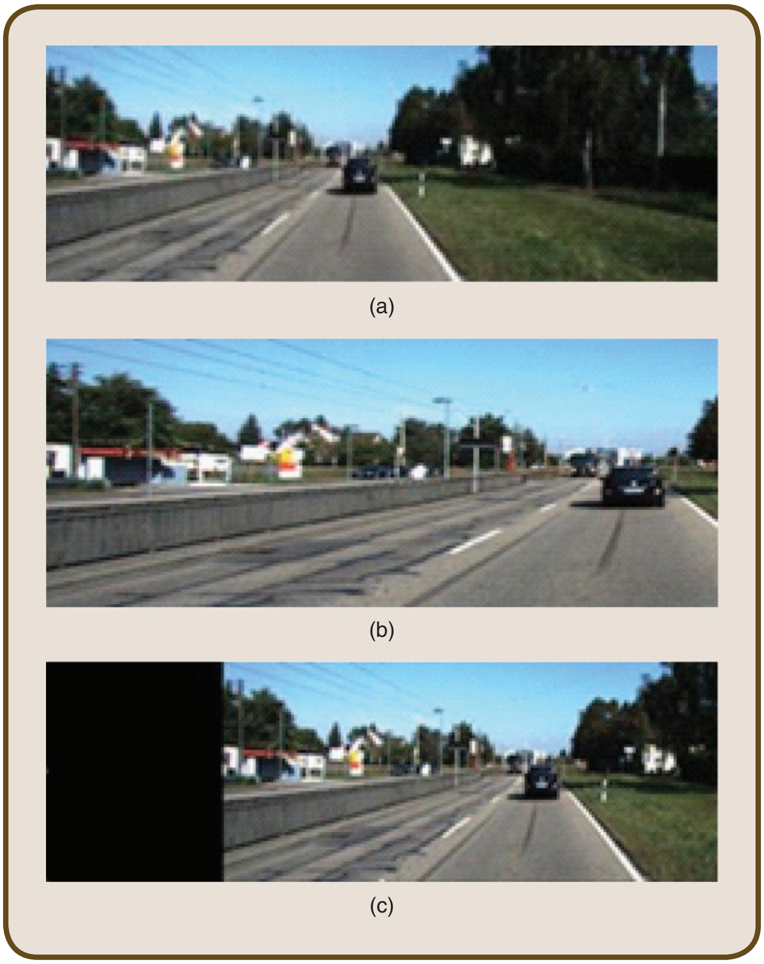


FIG 10 Data augmentation methods. (a) Noise adding, (b) cropping, and (c) displacement.

denotes the random gradient descent method. In theory, the larger the number of the input samples is, the truer the real distribution of the data is, but the large sample input in one iteration will require higher computing performance. On the other hand, the batch size can increase to a certain level (where its determined descent direction cannot be changed further) which will cause the network to fall into a sharp minimum, which will further make model have a poor generalization ability [39]. Taking all factors into consideration, in the experiment this article chose the batch size of 4. The initial value of learning rate in training was 10^{-4}, and the decay rate was set to 0.95 to prevent the gradient explosion during training. The IOU ratio used for matching the anchor boxes and the ground truth boxes was 0.5; thus, if the prediction boxes and real boxes had a fifty-percent coincidence, the prediction boxes were considered as positive samples; the IOU ratio selection of a non-maximum value suppression in the test was 0.9; namely, when the coincidence degree of the two detection boxes was more than ninety percent, the model filtered the box. To speed up the training, the current Glorot uniform distribution initialization method [40], also known as the Xavier uniform initialization, was used in the deep learning object detection. The parameters of this method were generated from the uniform distribution in the range [$-limit$, $limit$], where $limit$ was defined as $\sqrt{6/(fan\_in + fan\_out)}$, fan_in represented the number of input units of the weight tensor, and fan_out denoted the number of output units of the weight tensor.

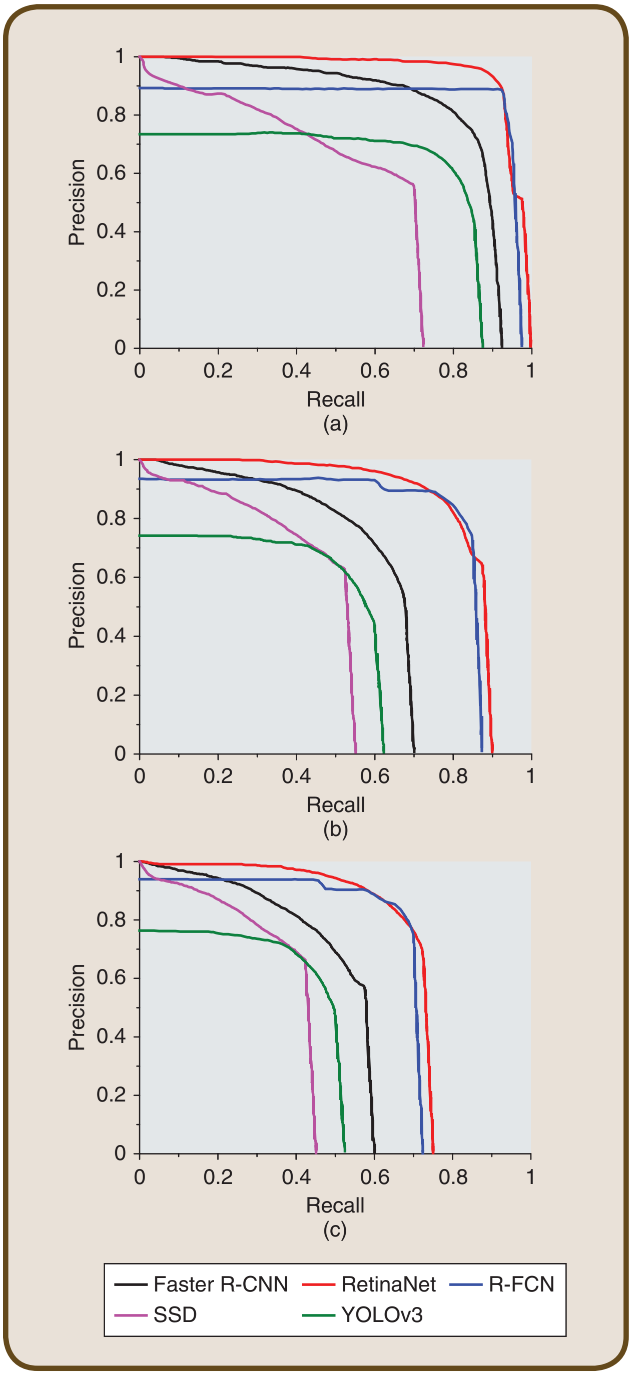


FIG 11 The PR curve of five deep models for vehicle detection. (a) Easy object, (b) moderate object, and (c) hard object.

D. Experimental Conditions and Results

In this experiment, this article divided the data set into two parts, 90% of the data were used for training, and 10% of the data were used for validation. The model training lasted for 100 epochs. The deep learning framework Tensorflow [41] was used for training. The test platform included the laptop equipped with a Core i5-7300HQ CPU and a GTX1050 GPU. The Adam gradient descent algorithm [42] was selected as an optimizer during training. The experimental results are obtained using the KITTI benchmark.

The data set of vehicles was divided into three detection-difficulty groups according to the ground truth bounding boxes size and occlusion degree: easy, moderate, and hard. If the bounding boxes size was larger than 40 pixels, a completely unshielded vehicle was considered to be an easy object, if the bounding boxes size was larger than 25 pixels but smaller than 40 pixels, a partially shielded vehicle was considered as a moderate object, and a vehicle with the bounding boxes size smaller than 25 pixels and an invisible vehicle that was difficult to see with the naked eye were considered as hard objects. The PR curve of vehicles with a different detection difficulty is shown in Fig. 11, where the average accuracy of detection and the number of detected picture frames per second are used as benchmarks to compare performances of five object detections models applied in vehicle detection, as shown in Table I.

As shown in Table I, the One-stage detection model RetinaNet achieves the highest detection accuracy for the detection of simple targets and difficult targets, and is also very close to the highest-rated R-FCN in the detection of moderate targets, but Retinanet has a huge detection speed advantage over R-FCN (there are still gaps compared with the two models of YOLOv3 and SSD, but the detection accuracy of these two types of models is very low). From Fig. 11, it can be found that the Recall value of the SSD is lower, From Fig. 11, it can be found that the Recall value of the SSD is low, and there is a large miss detection rate for the targets of three different detection difficulty, while the YOLOv3 is the opposite, with a higher Recall value and a lower Precision value. This means that YOLOv3 will have a large false detection rate, while R-FCN and RetinaNet have lower probability of missed detection and false detection, and RetinaNet will be more balanced than R-FCN.

The performance of each algorithm on some test samples is shown in Figure 12. Among them, this article selected three images with different detection difficulty to visually show the effect of the algorithm. From left to right the vehicle detection algorithm is followed by Faster R-CNN, R-FCN, RetinaNet, SSD and YOLOv3. It can be clearly seen from the figure that for the test samples with low detection difficulty, the five types of test models have good performance, and the difference is only the difference between the fit of the marquee and the target. The selection of RetinaNet and R-FCN will be more suitable for the target. For the target with moderate difficulty detection, the SSD model has begun to miss targets, and the farthest vehicle and the vehicle target with severely cut off at the lower right are not detected. For the difficult samples that are severely occluded, such as the target vehicle whose left side is close to the house, most of the detection models cannot be detected, and only the R-FCN detects the difficult target.

In addition to the testing using the KITTI test set, to investigate the generalization ability of each vehicle detection algorithm for real scenes, various vehicle detection algorithms were used to test other real road scenes during the day, night and rain, as shown in Fig. 13. It can be seen that R-FCN is very powerful for the generalization of real scenes, and can also have excellent detection effects in nighttime and rainy days. In addition, the single-stage model SSD also has a good generalization ability, and can detect most target vehicles in environments with poor lighting conditions.

E. Analysis of Results

According to the above experimental results, the two-stage vehicle detection algorithms generally had better detection accuracy than the one-stage vehicle detection algorithms, but the detection speed of one-stage detection algorithms was much better. Since the RetinaNet introduces the idea of focal loss and can get the detection accuracy far beyond the ordinary one-stage model, the performance of the RetinaNet on the KITTI test set exceeded some of the two-stage vehicle detection algorithms but at the cost of slow speed compared with other one-stage vehicle detection algorithms. Using the test samples with different detection difficulties, it is found that most of the vehicle detection algorithms performed well on the easy objects, but there was a large difference in the detection precision for the moderate and hard objects with the scale transformation and the occlusion impact, especially of the faster R-CNN, because there was no optimization of a multi-scale detection, so the detection accuracy decreased greatly. In consideration of the sensitivity of the position, the R-FCN showed strong adaptability to the occlusion vehicle. In addition, the RetinaNet also had the strong adaptability to multiscale changes by using the FPN and could also achieve good test results for occlusion and small-scale vehicle targets. At the same time, by observing the PR curve, it is found that the recall ratio (Recall) and the precision ratio (Precision) of the faster R-CNN, R-FCN, and RetinaNet were more balanced, while the SSD had a higher precision and lower recall rate, indicating that the SSD was more unchecked when detecting vehicles. On the contrary, the recall rate of the YOLOv3 was much higher than its precision, which means that the YOLOv3 had more misdetections. In the practical vehicle detection, the YOLOv3 with a high detection rate was more advantageous than the SSD in the security.

In order to further investigate the possibility of vehicle detection algorithms, this article used a test picture that was closer to the actual road condition. The R-FCN results showed that it had an excellent generalization ability. Even in the night, rain, and other complex conditions, the detection performance was excellent, but the generalization ability of the RetinaNet was very limited, and the detection precision on different distributed vehicle data was not better than of the other two kinds of one-stage detection algorithms. The results also showed that the faster YOLOv3 and SSD had better adaptability in the practical engineering application than the RetinaNet with excellent performance on the test set.

Table II. Performance of five deep models for vehicle detection on KITTI dataset.

Method	Average Precision (%)			FPS	Total Number
	Easy	Moderate	Hard		
Faster R-CNN	81.09	57.47	48.37	0.68	7518
R-FCN	81.24	79.49	66.01	0.31	7518
SSD	56.63	45.93	38.91	14.15	7518
YOLOv3	58.56	45.98	38.23	8.17	7518
RetinaNet	89.83	78.85	68.73	3.59	7518

According to the analysis of a series of experimental results made above, the accuracy, model complexity, sensitivity and scene adaptability of various algorithms are summarized and summarized in the Table III. This article will use high, medium, low as an indicator to measure the pros and cons of various types of algorithms.

In summary, although the detection speed of RetinaNet with higher detection accuracy is not as fast as YOLOv3 and SSD, it can still meet the real-time detection standard and is applied to the road driving environment perception link of actual vehicles. The R-FCN with excellent detection accuracy is limited by the low detection speed, which cannot meet the real-time demand for surrounding target detection when the vehicle is running. It is more suitable for detecting various obstacles around in low-speed scenes such as automatic parking. These scenes require higher algorithm generalization ability. And the YOLOv3 and SSD with extremely fast detection speed but poor detection accuracy is more suitable for deployment on various embedded devices due to their lower demand for computing resources. Such as unmanned logistics transport cars, etc., so that some mobile platforms with limited computing power can also perform real-time environment perception.

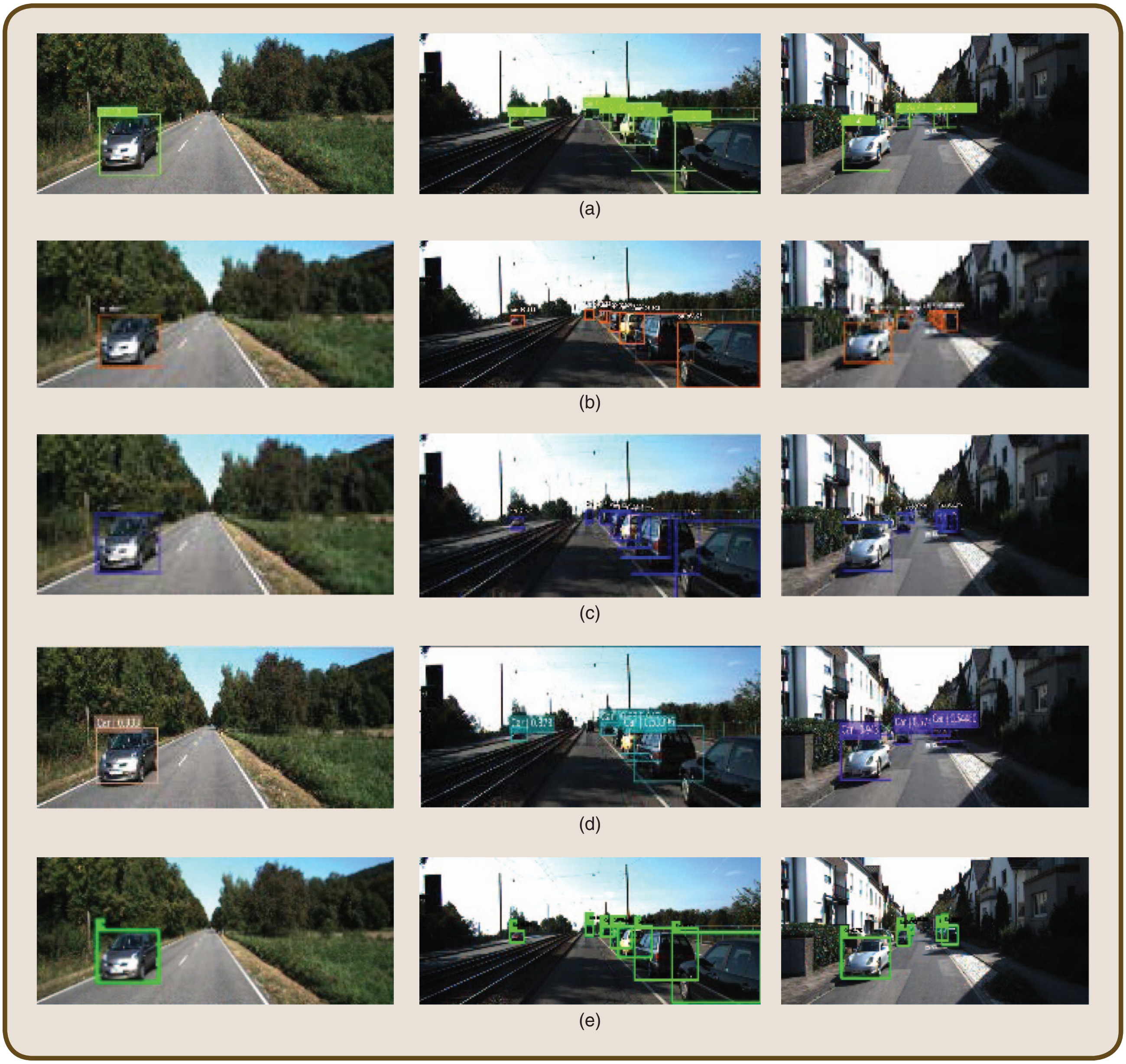
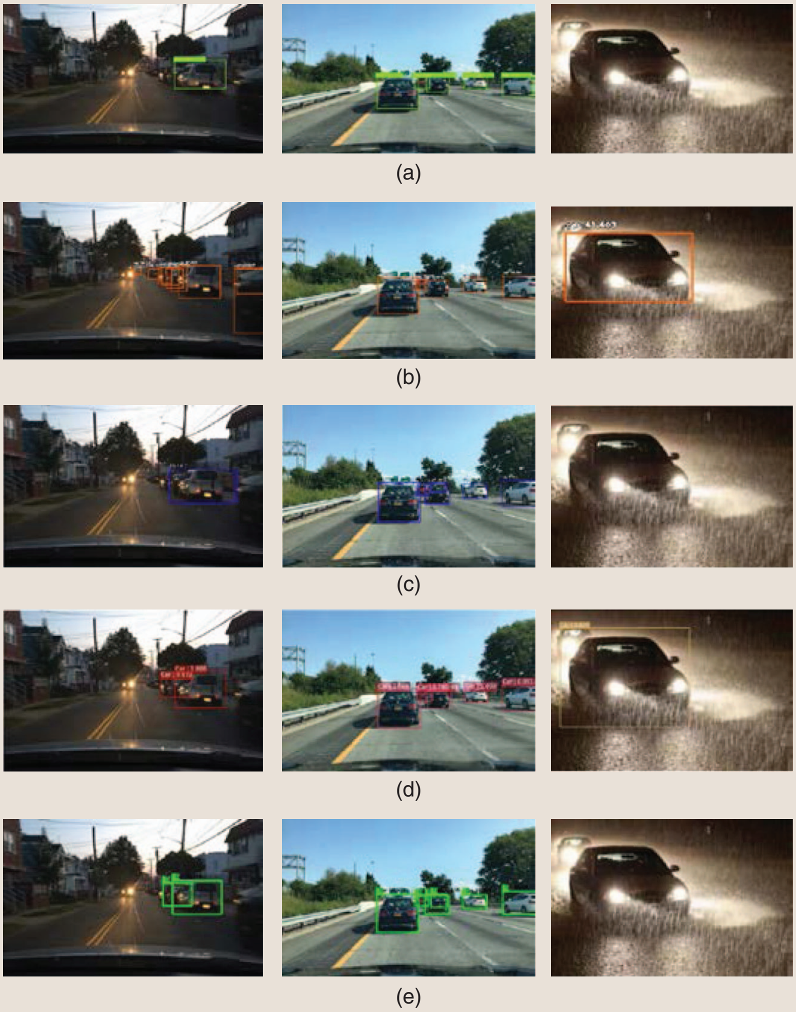


FIG 12 Performance of five deep model for vehicle detection tasks in three categories (from left to right column: easy, medium, and hard). (a) Faster R-CNN, (b) R-FCN, (c) RetinaNet, (d) SSD, and (e) YOLOv3.

F. Development Trend Prediction

Because of the particularity of road vehicle detection, the future vehicle detection algorithm based on deep learning can be deployed on all kinds of embedded systems like many traditional algorithms, but since the computing ability of embedded system is relatively smaller than of the computer platform, many models which can run on computers in real time may not perform well in embedded systems, so they may not detect a vehicle in real time. In order to solve this problem, a large number of researchers have made a series of efforts towards the further lighting of the detection models, and have proposed a series of algorithms for real-time performance, such as the lightweight model of RFCN named the Light Head R-CNN [43], which was proposed recently; in this model, before generating a score map, the partial convolution of the ResNet is replaced by the deep separable convolution, and the generated score maps are reduced. In this way, the model can achieve the real-time detection. In the light quantization model, the SSDLite [44], inherited from the SSD, the convolution operation in the detection head is replaced by the deep separable convolution, and the parameter is greatly reduced in this way.


(a)

(b)

(c)

(d)

(e)

FIG 13 Performance of five deep models on three different real road scenes. (a) Faster R-CNN, (b) R-FCN, (c) RetinaNet, (d) SSD, and (e) YOLOv3.

The combination with the image segmentation technology is another future development direction of the target detection algorithms. The existing object detection algorithms can be difficult to improve greatly under the original framework. Therefore, some researchers have begun to combine image segmentation with object detection, such as the Mask R-CNN [45], which combines the detection and instance segmentation training methods to improve the accuracy of segmentation and detection to achieve the pixel-level detection. Beside, visual saliency are also another possible useful tool to combine with deep network with the task of ROI generation [46], or object segmentation [47].

Table III. The pros and cons of several performances of various algorithms.

Method	Accuracy	Sensitivity	Specificity	Complexity
FasterR-CNN	Medium	Medium	Low	High
R-FCN	High	Low	High	High
SSD	Low	High	High	Low
YOLOv3	Low	High	Medium	Low
RetinaNet	High	Medium	Low	Medium

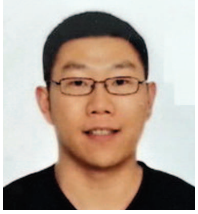
V. Discussion

Through the experimental evaluation in the previous section, It can be seen that most of the two-stage models have better detection accuracy than the one-stage object detection model, as shown in Table I, but RetinaNet has the outstanding detection accuracy, this is because the RetinaNet model uses a special loss function named FocalLoss, which can effectively reduce the weight of easily categorized samples, so that the model is more focused on difficult-to-classify samples during training. With RPN in the two-stage model, RetinaNet is very suitable for applications that require detection accuracy and have certain requirements for real-time performance of the algorithm, such as applied to intelligent vehicles for road targets detection. In addition, through comparative experiments, this paper also finds that although SSD is an old deep learning target detection algorithm, it still has a huge advantage in real-time compared to the updated YOLOv3. It can be found through analysis that SSD does not use the idea of FPN, and the detection model apart from the feature extract net is also relatively simple, which greatly reduces the complexity of the model. It is also because of its lightweight structure, which has low requirements on computing power. This work considers the SSD model to be very suitable for various types of mobile platforms, such as small unmanned aerial vehicles and logistics trolleys, instead of traditional target detection algorithms such as sliding window method. On the other hand, this paper tests various algorithms under three types of bad detection conditions, as shown in Figure 13. It is unexpectedly found that the SSD model has excellent generalization ability. This paper thinks that this is due to the relatively simple SSD. The model construction is less prone to over-fitting of a certain data set, and has strong detection robustness, which proves the feasibility of SSD applied in practical engineering. Researchers who are heavily committed to applying deep learning object detection algorithms to intelligent vehicles can get inspiration from this paper, for example, when the researchers try to use the complex net to improve the detection performance, modifying the loss function would be a better choice instead of add a lot of layer in the model. In addition, it can be found that how to accurately detect small targets without adding FPN network structure (FPN structure will greatly increase the complexity of the model) is still a scientific problem to be solved in future deep learning.

VI. Conclusion

In this paper, five independent deep learning target detection algorithms are compared in road vehicle detection. Each model was trained using the same KITTI public dataset assigned to the same training scale. Three indicators were used to compare the comprehensive performance of the algorithms: (1) the recall rate and precision rate on the KITTI test set; (2) the average precision on the KITTI test set; (3) the fps. In addition, this work also uses various models to detect vehicles in a real road scene to make an intuitive judgment on the applicability of each target detection algorithm for road vehicle detection. In general, as the smart driving is getting closer and closer to us, the road vehicle detection based on deep learning is still a subject worthy of studying. The main challenge is to balance the real-time performance and accuracy. However, more in-depth research is needed for solving this challenge.

About the Authors



Hai Wang (M'17) received the B.S., M.S. and Ph.D. degree all from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, respectively. In 2012, he joined the School of Automotive and Traffic Engineering in Jiangsu University, now he is an associate professor. His research interests include computer vision, intelligent transportation systems and intelligent vehicles. He has published more than 50 papers in the field of machine vision based environment sensing for intelligent vehicles.

Yijie Yu received the B.S. degree from Zhejiang Institute of Technology, Hangzhou, China. He is now a master student at School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China. His research interests include computer vision, deep learning, intelligent vehicles.

Yingfeng Cai (M'17) received the B.S., M.S. and Ph.D. degree all from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, respectively. In 2013, she joined the Automotive Engineering Research Institute in Jiangsu University, now she is a professor. Her research interests include computer vision, intelligent transportation systems and intelligent automobiles.

Xiaobo Chen received the Ph.D. degree in Nanjing University of Science and Technology, Nanjing, China, in 2013. In 2013, he joined the Automotive Engineering Research Institute in Jiangsu University, now he is an associate professor. His research interest include machine learning, intelligent transport systems.

Long Chen received his Ph.D. degree in Vehicle Engineering from Jiangsu University, Zhenjiang, China, in 2002. In 2010, he joined the Automotive Engineering Research Institute in Jiangsu University, now he is a professor. His research interests include intelligent automobiles and vehicle control systems.

Qingchao Liu received the Ph.D. degree from the School of Transportation Engineering, Southeast University, Nanjing, China. In 2015, he joined the Automotive Engineering Research Institute in Jiangsu University, now he is an assistant professor. His research interests include intelligent vehicles intelligent transport systems.

References

[1] J. W. Runge, "National highway traffic safety administration," *Public Health Rep.*, vol. 102, no. 6, pp. 667–668, 1987.

[2] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, 2006.

[3] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, D. Li, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, 2018.

[4] M. M. Trivedi, T. Gandhi, and J. Mccall, "Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 108–120, 2007.

[5] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2015, pp. 91–99.

[6] J. Dai, Y. Li, K. He, J. Sun. "R-FCN: object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[7] W. Liu et al. "SSD: single shot multibox detector," in *Proc. European Conf. Computer Vision*. New York: Springer International Publishing, 2016, pp. 21–37.

[8] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Computer Vision*, 2017, pp. 2999–3007.

[9] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," arXiv Preprint, arXiv: 1804.02767, 2018.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 2012, pp. 3354–3361.

[11] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2019–2034, 2006.

[12] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *Proc. 10th IEEE Int. Conf. Computer Vision*, 2005, pp. 1619–1626.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, vol. 1, no. 12, pp. 886–893, 2005.

[14] G. Zhang, X. Huang, S. Z. Li, Y. Wang, X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. Chinese Conf. Advances Biometric Person Authentication*. New York: Springer-Verlag, 2004, pp. 179–186.

[15] R. Capparuccia, D. L. Renato, and E. Marchitto, "Integrating support vector machines and neural networks," *Neural. Netw.*, vol. 20, no. 5, pp. 590–597, 2007.

[16] W. Xuezhi, F. Wei, and Z. Yuhui, "A vehicle identification algorithm based on Haar-like features and improved AdaBoost classifier," *Chin. J. Electron.*, vol. 39, no. 5, pp. 1121–1126, 2011.

[17] C. Leistner, P. M. Roth, H. Grabner, H. Bischof, A. Starzacher, B. Rinner, "Visual on-line learning in distributed camera networks," in *Proc. ACM/IEEE Int. Conf. Distributed Smart Cameras*. Piscataway, NJ: IEEE, 2008, pp. 1–10.

[18] A. J. Joshi and F. Porikli, "Scene-adaptive human detection with incremental active learning," in *Proc. Int. Conf. Pattern Recognition*. Piscataway, NJ: IEEE, 2010, pp. 2760–2763.

[19] A. Takeuchi, S. Mita, and D. Mcallester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," in *Proc. IEEE Intelligent Vehicles Symp.*, Piscataway, NJ: IEEE, 2010, pp. 1014–1021.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proc. 2001 IEEE Computer Society Conf.*, Piscataway, NJ: IEEE, 2003, vol. 1, pp. I-511–I-518.

[21] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2019–2034, 2006.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Information Processing Systems*. Red Hook, NY: Curran Associates Inc. 2012, pp. 1097–1105.

[23] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[24] R Girshick, J Donahue, T Darrell, J Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[25] J. R. R. Uijlings, V. D. Sande, T. Gevers, A. W. M. Smeulders. "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[26] R. Girshick, "Fast r-cnn," *Computer Sci.*, arXiv Preprint, arXiv: 1504.08083, 2015.

[27] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 2016, pp. 779–788.

[28] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," arXiv Preprint, arXiv: 1612.08242, 2016, pp. 6517–6525.

[29] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2117–2125.

[30] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 2016, pp. 770–778.

[31] K. He, X. Zhang, S. Ren, J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 2015, pp. 3431–3440.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Sci.*, arXiv Preprint, arXiv: 1409.1556, 2014.

[34] T. Y. Lin et al., "Microsoft COCO: common objects in context," in *Proc. European Conf. Computer Vision*, Cham: Springer, 2014, vol. 8693, pp. 740–755.

[35] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2015, pp. 1–9.

[36] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[37] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining,"in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[38] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *J. Roy. Stat. Soc.*, vol. 28, no. 1, pp. 100–108, 1979.

[39] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. Tang, "On large-batch training for deep learning: generalization gap and sharp minima," arXiv Preprint, arXiv: 1609.04836, 2017.

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[41] M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous distributed systems," arXiv Preprint, arXiv: 1603.04467, 2015.

[42] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Comput. Sci.*, arXiv Preprint, arXiv: 1412.6980, 2014.

[43] L. Zeming, C. Peng, G. Yu, Y. Deng, J. Sun, "Light-head R-CNN: in defense of two-stage object detector," arXiv Preprint, arXiv: 1711.07264, 2017.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[45] K. He, G. Gkioxari, P. Dollar, R. Girshick, "Mask r-cnn," in *Proc. IEEE Transactions Pattern Analysis and Machine Intelligence*, 2017, vol. 99, pp. 2961–2969.

[46] Y. Cai, Z. Liu, H. Wang, X. Sun, "Saliency-based pedestrian detection in far infrared images," *IEEE Access.*, vol. 5, pp. 5013–5019, 2017.

[47] H. Wang, L. Dai, Y. Cai, X. Sun, L. Chen, "Salient object detection based on multi-scale contrast," *Neural Netw.*, vol. 101, pp. 47–56, 2018.