IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 20, 2020, accepted May 31, 2020, date of publication June 10, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001279

Data-Driven Based Tiny-YOLOv3 Method for Front Vehicle Detection Inducing SPP-Net

XIAOLAN WANG, SHUO WANG, JIAQI CAO, AND YANSONG WANG, (Member, IEEE)
School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Xiaolan Wang (jlu_wangxiaolan@aliyun.com)

This work was supported in part by the Project of National Natural Science Foundation of China under Grant 51675324 and Grant 51175320, in part by the Project of Automotive Industry Science and Technology Development Foundation of Shanghai under Grant 1523, and in part by the Program for Professor of Special Appointment (Eastern Scholar), Shanghai Institutions of Higher Learning, China.

ABSTRACT In order to solve the problem of low recognition rate and low real-time performance of vehicle detection in complex road environment, a data-driven forward vehicle detection algorithm based on improved tiny-YOLOv3 is proposed. Based on tiny-YOLOv3, the context feature information is combined to increase the two scale detections of tiny-YOLOv3 to three. The spatial pyramid pooling (SPP) module is added to increase the number of feature channels to improve the network feature extraction ability. According to the dense arrangement of vehicles on the horizontal axis in the road image ahead, we change the grid size of tiny-YOLOv3 and increase the number of candidate boxes on the horizontal axis. In addition, combined with the characteristics of the vehicle size in the road image ahead, K-means clustering method is used to select the appropriate number and size of target candidate boxes. We obtain the optimal detection model by multi-scale training of the improved network. The experimental results show that the average accuracy of the improved algorithm on the KITTI datasets is 91.03%, which is 7.12% higher than that of tiny-YOLOv3. And the detection speed of improved network is 144 frames/s, which meets the real-time requirements.

INDEX TERMS Data-driven, convolutional neural network, k-means, spatial pyramid pooling, vehicle detection.

I. INTRODUCTION

Intelligent vehicle will be the inevitable trend of the future development of automobile industry. With the development of artificial intelligence, autonomous driving has attracted extensive attention of researchers [1]. As an important premise of automatic driving, the perception of the road environment ahead has become the research focus in the field of intelligent vehicles. To ensure safe driving, intelligent vehicles should deeply understand the vehicle behavior in front of them according to the rich dynamic target parameters [2]. Accurate and real-time detection of vehicles in front can be used to determine the automatic driving path of the vehicle, which is crucial in environmental perception [3]. Vehicle detection methods mainly include traditional detection algorithm and deep learning method.

Traditional vehicle detection methods include optical flow method, background subtraction method, and detection method based on appearance features. Horn and Schunck [4] proposed a determining optical flow method to detect moving objects. However, the target detection based on optical flow method is easily affected by noise and light source changes, resulting in poor robustness in different scenes. Background subtraction algorithm is only suitable for monitoring video with static background, and it can't identify vehicles with static or slow-moving speed [5]. The vehicle detection method based on appearance features mainly detects the edge, symmetry and color [6]–[8] of the vehicle. For example, van Leeuwen and Groen [9] proposed a vehicle detection method based on the characteristics of vehicle shadow and vehicle symmetry. However, when the moving target is deformed or the scene perspective changes, the effect of feature representation will become worse, which will make the detection effect worse.

The traditional vehicle detection algorithm can be divided into three steps: Firstly, the region of interest is selected, and the entire image is scanned through a multi-scale sliding window [10] according to the position and size characteristics of the vehicle in the picture. Although it can get more accurate vehicle position and size, this method requires a

The associate editor coordinating the review of this manuscript and approving it for publication was Guangdong Tian.

huge amount of calculation and will produce redundant marking box because of treating each area indiscriminately. Secondly, features are extracted from the candidate regions, and commonly used are manual features such as HOG [11], har-like [12] and LBP [13]. Due to the influence of target occlusion, illumination changes, and background interference, the artificially designed image features are poorly robust and it is difficult to express the target features in all cases. Reference [14] shows that a vehicle detection method based on HOG feature is proposed, which has 88% detection accuracy, but it is easy to cause false detection and poor robustness. Finally, according to ANN neural networks [15], SVM [16], Adaboost [17], and other methods to classify, and complete vehicle detection. However, the traditional vehicle detection method has poor adaptability to the change of environment and vehicle target, which cannot meet the requirements of unmanned driving.

With the development of deep learning and GPU, target detection technology based on deep learning [18]–[20] is more and more widely used, which has better detection effect than traditional methods. The basic idea of using convolutional neural network for vehicle detection is to analyze the underlying features and corresponding target tags of images by means of supervised learning. Instead of using the characteristics of artificial design, a group of network weights with the minimum loss function are obtained. The weight of the trained network is loaded into the network, and the vehicle target in the image is identified by the forward reasoning of the network. The vehicle detection algorithm based on convolutional neural network has better robustness. It can overcome the influence of light change, shadow noise, and obstacle occlusion, and becomes the research trend of vehicle detection field. This method can be divided into two categories, one is based on the two-step method of regional recommendation. The R-CNN [21] generates candidate regions by Region recommendation, then uses CNN to extract features in each candidate region and sends the features to the SVM. The classifier determines the target category, and finally R-CNN uses linear ridge regression to adjust the position of candidate regions. In recent years, this method has been improved continuously. SPP-net [22], Fast R-CNN [23], Faster R-CNN [24], Mask R-CNN [25], and other target detection methods are proposed, which have achieved better detection results. However, due to the complexity of the two-step network structure and poor real-time performance, it is difficult to realize the application [26].

The other type is a one-step approach based on regression methods. Representatives include YOLO [27], SSD [28] etc. The YOLO (You Only Look Once) algorithm improves the accuracy of obtaining local information of image. The false detection rate of background is reduced and the detection speed is accelerated. Its lightweight version, tiny-YOLO [29], has achieved a detection speed of 155 frames/s. However, the accuracy is relatively low and it is not good at detecting small objects. To solve the problem of low detection accuracy and recall rate, Redom and others improve YOLO by regularization and dimensional clustering, and propose YOLOv2 [30]. The mAP (mean Average Precision) is 76.8% on the VOC2007 datasets, and the test speed is 67 frames/s. In April 2018, the third improved version YOLOv3 [31] was published, and the mAP on the COCO datasets was increased from 44.0% to 57.9% of YOLOv2, achieving high accuracy under the premise of guaranteed speed. At the same time, due to the deepening of the network and the increase of the amount of calculation, the requirements of hardware were higher and higher. For the convenience of deployment, the corresponding convolutional network is simplified, such as Mobile-Net [32], tiny-YOLO, and tiny-SSD [33] and so on. There are fewer convolution layers in these networks. The detection accuracy is sacrificed to a certain extent, but the detection speed is faster.

To achieve accurate and real-time detection of front vehicles in complex environment, the first part of this paper reviews the research status of vehicle detection methods and puts forward problems. In the second part, the principle of the series of YOLO algorithms is described. In the third part, a vehicle detection algorithm based on improved tiny-YOLOv3 is proposed. Based on the tiny-YOLOv3 network, the vehicle in the road image ahead is taken as the target. To improve the detection ability of small targets, the tiny-YOLOv3 prediction layer is improved. The detection scale is increased by combining the low-level and high-level feature map of context information fusion, and the spatial pyramid pooling is introduced to increase the number of feature map channels to retain more target information. At the same time, to ensure the real-time performance and improve the detection accuracy to meet the actual needs, the grid size, the selection of candidate boxes, and network training are improved. The fourth part we compare the improved tiny-YOLOv3 with the previous series of YOLO algorithms and analyze the experimental results, which verifies that the proposed method has better detection ability.

II. THE PRINCIPLE OF TINY-YOLOV3 ALGORITHM

YOLO is an end-to-end target detection algorithm, which transforms the problem of target detection into a regression problem. The classification task and location task are unified in a network. The location and category probability of candidate frame are predicted directly, which meets the real-time requirements.

The YOLO detection model is shown in Fig. 1. The original image is divided into S×S cells after zooming. If the cell has the center of the object to be detected, the location information and category information of the object to be detected are predicted by the cell. Each cell predicts the conditional probability of categories C, bounding boxes B and their confidence scores. Each bounding box predicts information, including coordinates (x, y), width w and height h of the target and confidence, which are recorded as t_x, t_y, t_w, t_h, and obj_conf. The confidence formula is:

$$obj\_conf = P_r(obj) \times IOU_{pred}^{truth} \quad (1)$$

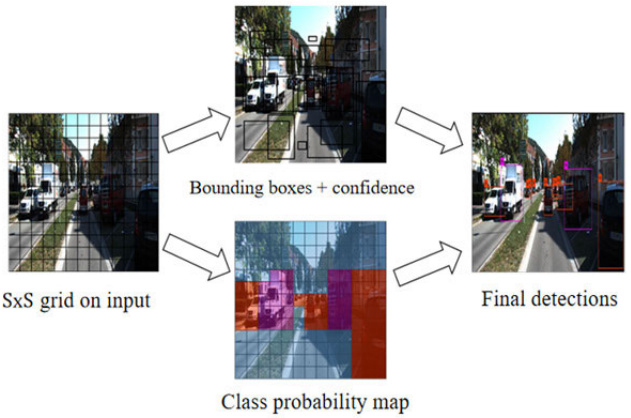

FIGURE 1. The detection model of YOLO.

where IOU_{pred}^{truth} is the intersection ratio of prediction box and real value, which is used to judge the accuracy of target position. $P_r(obj)$ is whether there is a target in the prediction bounding box corresponding to the cell. If there is no target, it means 0. If there is one, it means 1. The confidence formula reflects whether the cell contains the target or not and the accuracy when the prediction bounding box contains the object. (c_x, c_y) is the horizontal and vertical offset of the cell of the target center to be detected from the upper left corner of the image. The candidate box has a width of p_w, and a height of p_h. The coordinate calculation formula of the inspection bounding box are as following.

$$b_x = \sigma(t_x) + c_x \tag{2}$$

$$b_y = \sigma(t_y) + c_y \tag{3}$$

$$b_w = p_w e^{t_w} \tag{4}$$

$$b_h = p_h e^{t_h} \tag{5}$$

where t_x, t_y, t_w, and t_h represent the predicted values of the center, width and height of the detection box on the candidate box. $\sigma(\bullet)$ is the sigmoid activation function, which is used to limit the center point of the detection box within the grid. b_x, b_y, b_w, *and* b_h are the horizontal, vertical, width and height of the detection box center. When more than one bounding box detects the same target, YOLO uses non maximum suppression method to filter the bounding box with lower threshold to get the best target prediction box.

Although YOLO has faster detection speed than Faster R-CNN, its detection accuracy is lower. To solve this problem, YOLOv2 improves the network structure and replaces the full connection layer in the output with convolutional layer. YOLOv2 also introduces batch normalization, high-resolution classifier, dimensional clustering, fine-grained features, multi-scale training, and other methods. However, there are still shortcomings of poor detection of small targets. YOLOv3 is based on YOLO and YOLOv2. Using the idea of deep residual network for reference, a residual module is built between convolution layers. The jump connection is set, and a deeper convolution neural network Darknet-53 is designed. The complete network structure has 106 layers, which has better feature extraction effect and improves the positioning and classification accuracy of target detection. Tiny-YOLOv3 is a simplified target detection algorithm based on the version of YOLOv3, which reduces the amount of calculation and greatly improves the speed. At the same time, this method reduces the requirements of hardware and increases the possibility of application. As shown in Table 1, the backbone network consists of 7-layer convolutional and 6-layer pooling.

TABLE 1. Tiny-YOLOv3 backbone network structure.

Type	Filter	Size	output
Convolutional	16	3 × 3/2	416 × 416 × 16
Pooling		2 × 2/2	208 × 208 × 16
Convolutional	32	3 × 3/2	208 × 208 × 32
Pooling		2 × 2/2	104 × 104 × 32
Convolutional	64	3 × 3/2	104 × 104 × 64
Pooling		2 × 2/2	52 × 52 × 64
Convolutional	128	3 × 3/2	52 × 52 × 128
Pooling		2 × 2/2	26 × 26 × 128
Convolutional	256	3 × 3/2	26 × 26 × 256
Pooling		2 × 2/2	13 × 13 × 256
Convolutional	512	3 × 3/2	13 × 13 × 512
Pooling		2 × 2/1	13 × 13 × 512
Convolutional	1024	3 × 3/2	13 × 13 × 1024

III. THE PRINCIPLE OF IMPROVED DETECTION ALGORITHM

A. SPP-NET

SPP-net is a kind of pyramid network, which is connected to Gaussian pyramid pooling layer after the last convolution layer. Pyramid pooling layer can transform any size feature map into fixed size feature vector, then match with full connection layer. With this method, any size image can be used as the input of neural network, and a fixed size output can be generated [34], [35]. SPP-net completes multi-level feature extraction through spatial pyramid pooling, enhances the robustness of the network and improves the detection accuracy and speed. In the same way, SPP-net extracts characteristic graphs of different sensory field sizes by pooling layers of different sizes to combine global and sub-regional information. Furthermore, the number of channels in the feature graph is widened to provide effective global context information. Therefore, it has a stronger ability of detail feature description, and improves the detection accuracy of different types of targets.

In this paper, two sets of SPP-net spatial pyramid pooling modules are integrated, and SPP-net is adjusted to introduce four pooling layers, whose dimensions are 1 × 1, 2 × 2, 3 × 3, 4 × 4, and 1 × 1, 5 × 5, 9 × 9, 13 × 13. The specific structure is shown in the Fig. 2.

B. CONSTRUCTION OF FRONT VEHICLE DETECTION MODEL

Since the size and proportion of the vehicles in the road image in front are not fixed, it is easy to miss or judge

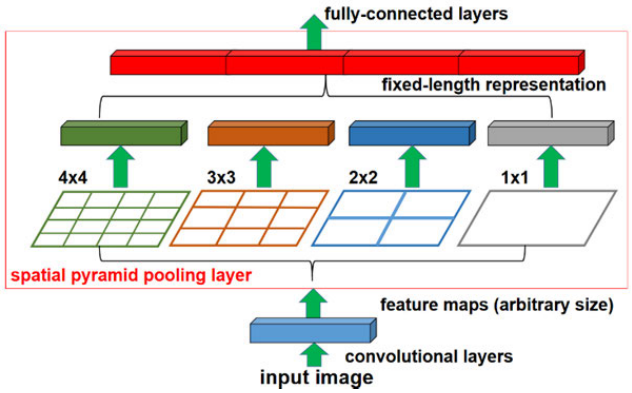

FIGURE 2. Adjusted SPP-net structure with dimensions of 1 × 1, 2 × 2, 3 × 3, 4 × 4.

the vehicle as another type of target object when the target is far away or the vehicles overlap each other. The tiny-YOLOv3 network is a simplified network for multi-category target detection. It has achieved good real-time performance in detecting road vehicles in front, but the number of layers in the network is small, and it is difficult to extract the vehicle target features. The position and probability of the target object are predicted only in the high-level feature map of two scales, so there are problems such as poor positioning accuracy for small targets, low target recognition rate of the vehicle, false detection or repeated detection. In order to further enhance the detection capability of the target, the tiny-YOLOv3 network was improved.

In the basic neural network, the amount of feature information obtained by the target is different at the final output because the size of the target in the image is different. Low-level and large-scale feature maps have high resolution and they can describe more accurate position information, but less semantic information; high-level feature maps contain richer semantic information, but the location information of the target points is sketchy. Therefore, the shallower convolutional layer can well represent the small-sized target, and the feature map is representative of the small target's position. That is, the large-scale feature map corresponds to the small target, and the deeper convolution layer has better features. In short, the convolutional layer features of different scales are selected according to different target sizes, and the features of high and low layers are integrated to obtain more semantic information, so as to predict targets, which can have better adaptability to targets of different sizes.

In this paper, based on the tiny-YOLOv3 network, the low-level features are fused with the high-level features through the upper sampling. One detection scale is added, and three feature layers of different scales are used for detection. At the same time, the spatial pyramid pooling module of SPP-net is integrated, and the grid size is changed. The vehicle detection model in complex environment is proposed, and more suitable candidate frames are allocated to small target vehicles in the increased scale. We call this network structure PPt-YOLOv3. The specific network structure of the model is shown in Fig. 3.

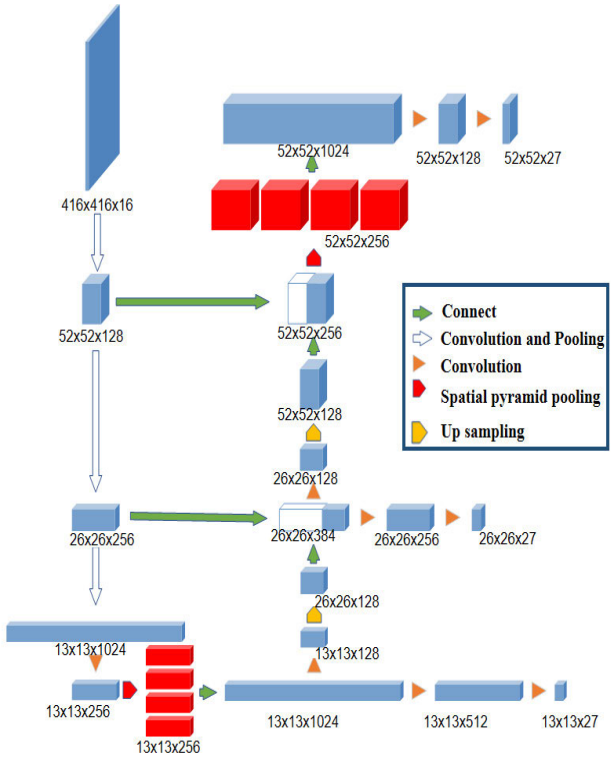

FIGURE 3. The structure of PPt-YOLOv3 network caption.

When the size of the input detection image is 416 × 416, the characteristic images of 52 × 52 × 128, 26 × 26 × 256, and 13 × 13 × 1024 are obtained after a series of convolution pooling. After that, the feature map of 13 × 13 × 256 is obtained by one-time convolution, at this time we access it to SPP-net. In the feature map, four pooling modules of 1 × 1, 5 × 5, 9 × 9, and 13×13 are introduced. The maximum pooling is used to retain more target texture information in the corresponding scale feature map as much as possible. Then, the feature map of 13×13 × 1024 is obtained by splicing. In order to help the network learn fine-grained features, on the original tiny-YOLOv3 network structure, the feature maps of 13 × 13 × 128 and 26×26 × 128 are sampled twice to obtain the feature maps of 26 × 26×128 and 52×52 × 128. Combined with the context information the feature maps with 384 and 256 channels are obtained. At this time, the second SPP-net, and four pooling modules with the dimensions of 1 × 1, 2 × 2, 3 × 3, 4 × 4 are introduced. After pooling, the characteristic diagram of 52×52 × 1024 is obtained by splicing. After another convolution calculation, the characteristic figure of 52×52 × 128 is obtained. The whole network finally obtains the feature maps of 13 × 13, 26×26, 52 × 52. The feature maps of three scales are respectively predicted and output through a convolutional layer, whose channel is $(5 + 4) \times 3 = 27$.

C. GRID SIZE

In the YOLO detection algorithm, the images are divided into S×S networks, and the horizontal vertical detection weights are the same. When detecting the vehicles on the road ahead, it can be found that the vehicle targets are closely arranged in the horizontal direction and sparsely distributed in the vertical direction in the image, and the original candidate frame distribution rules are difficult to apply. To solve this problem, we change the length width ratio of the network model input. As shown in Fig. 4, we increase the number of horizontal grids and the number of candidate frames in the horizontal direction, and refine the grid to ensure that the vehicle center falls into the correct Cell. The original network input image size is 416 × 416. In order to avoid the influence of input image resolution on the network, we select 768 × 384 resolution image as the network input, that is, the number of meshes is 24 × 12, which can have better horizontal feature extraction effect. In a word, we improve the positioning accuracy of the model and further increase the detection accuracy by changing the grid size.

FIGURE 4. Grid scale.

D. SELECTION OF CANDIDATE BOX AND NETWOEK TRAINING

In this paper, the detection scale of tiny-YOLOv3 is increased to three. The feature layers of different scales need to allocate the corresponding size of candidate frame to play the advantages and improve the detection ability. Taking the Mean Intersection over Union (MIOU) as the evaluation standard, K-means clustering method is used to get the dimension of the candidate frame for the dimension of the training set of the vehicle in front. The larger the value of K is, the more classes are clustered, the more accurate the classification of candidate box size is. However, the larger the value of K is, the more the total number of candidate frames in the network is, which means that the more computation is, the more complex the model is.

We choose 768 × 384 as the model input size, and use the incremental method to select the K value. The relationship between MIOU and the value of K is shown in Fig. 5. As the number of K increases from 1 to 15, the MIOU increases gradually. When K is greater than 8, the MIOU value rises slowly and is basically stable. Considering the amount of computation of the network, and the improved

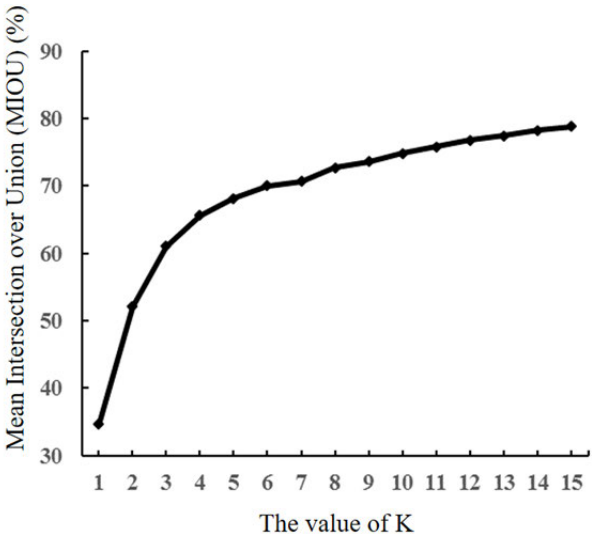

FIGURE 5. Relationship between the number of cluster centers and the MIOU.

method of prediction on three scales in the tiny-YOLOv3 network, the clustering result of K=9 is finally adopted. The dimensions of the 9 candidate boxes are respectively (20, 25), (35, 39), (66, 46), (50, 71), (92, 81), (141, 116), (99, 173), (199, 183), and (228, 325). Each cell on each scale predicts three check boxes with three candidate boxes. In other words, candidate boxes of size (99, 173), (199, 183), and (228, 325) are assigned to feature graphs with a sampling resolution of 13×13 at 32 times, which are used to detect large-size vehicles. The (50, 71), (92, 81), and (141, 116) size candidate boxes are allocated to the feature map with 16 times down sampling resolution of 26 × 26. A candidate boxes of size (20, 25), (35, 39), and (66, 46) is allocated to the feature map with 8 times lower sampling resolution of 52 ×52 to detect the small target vehicle in the distance.

Based on the open source deep learning framework Darknet, the improved tiny-YOLOv3 model combines clustering analysis and multi-scale training methods to train vehicle detectors. The initial learning rate of the model during training is set to 0.001. After 25 000 and 35 000 iterations, the learning rate is multiplied by 0.1. The momentum coefficient is 0.9, and the weight attenuation coefficient is 0.0010. The maximum number of iterations is 50,000. In the training, image random adjustment of exposure, saturation, tone, and other methods to expand the data. In addition, during the training, the multi-scale training strategy is adopted to enhance the robustness of images of different sizes. Each 10 batches of training randomly select new image sizes for training, so that the model has better detection effect for images of different sizes.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATA SET

The vehicle detection data in this paper are from the KITTI datasets. The KITTI datasets contain real-world image data

from scenes such as urban, rural, and highways, with up to 15 vehicles and 30 pedestrians per image, as well as varying degrees of occlusion and truncation images including intense lighting and blurred images, insufficient lighting, background noise, etc. According to the actual application scenario, we process the original 8 types of label information of the KITTI datasets, and retain the 4 category labels required for the experiment, namely: Van, Car, Truck, and Tram, and selects 7481 images in the datasets as experimental data. According to the experimental requirements, it is marked as PASCAL VOC2007 data set format, 80% of which are used as training set and 20% as verification set.

B. EXPERIMENTAL PLATFORM

The experimental platform configuration in this paper is shown in Table 2.

TABLE 2. The hardware and software configuration of experimental platform.

Designation	Configuration
Operating system	Windows 10
CPU/GHZ	Inter core i5-8400/2.80GHz
Memory/GB	16
GPU	1070ti
GPU accelerated Library	CUDA9.0 CUDNN7.0
Data processing	Opencv
Deep Learning Framework	Darknet

C. RESULTS AND ANALYSIS

In order to detect the rapidity, accuracy and robustness of the algorithm, we adopt four indexes: mean Average Precision (mAP), recall R, Intersection over union (IOU) and detection Precision P. The calculation formula of some measurement indexes are as follows:

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

$$IOU = \frac{A \cap B}{A \cup B} \tag{8}$$

where TP means true positive, FN means false negative, FP means false positive, TN means true negative.

Table 3 shows the experimental results of YOLOv2, tiny-YOLOv2, tiny-YOLOv3 and PPt-YOLOv3 proposed in this paper. All of these methods are trained and tested using the KITTI datasets. It can be seen from the table that the PPt-YOLOv3 obtains 91.03% mAP, and the precision of the network is improved by 7.12% compared with that of the tiny-YOLOv3 network, and the speed is reduced by 44 frames/s. The convolutional layer number of tiny-YOLOv2 and tiny-YOLOv3 are relatively small, and the vehicle feature extraction is insufficient. PPt-YOLOv3 network solves this problem by adding detection layer and spatial pyramid pooling, so it has excellent expression ability for vehicle features; while ensuring the accuracy, the detection speed is faster because fewer convolution layers are used.

TABLE 3. The test results of different methods on KITTI test set.

Method	Detection of frames	Precision/%				
		mAP	Van	Car	Truck	Tram
YOLOv2	114	69.48	60.81	66.45	69.57	81.07
tiny-YOLOv2	286	67.53	64.89	66.32	65.15	73.75
tiny-YOLOv3	188	83.91	85.53	80.49	88.17	81.44
PPt-YOLOv3	**144**	**91.03**	**90.62**	**90.29**	**90.73**	**92.56**

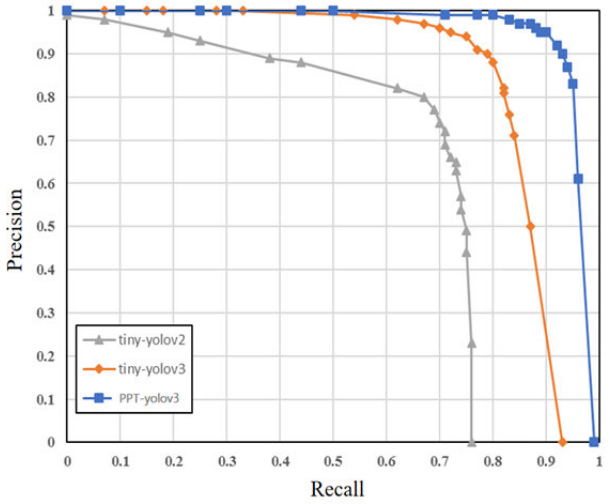

FIGURE 6. PR curve of different methods on KITTI.

Fig. 6 shows the PR curves of the three methods on the KITTI, in particular, where the precision is the average of the four vehicle types. It can be seen from the figure that by comparing the area under the curve, the method in this paper obtains the best performance, which shows that the improved network structure in this paper is effective. In addition, it can be seen from table 3 that although the speed of PPt-YOLOv3 is not the highest, the speed 144 frames/s is far beyond the requirements of real-time detection.

Mean Intersection over union (MIOU) is used as the index for evaluation to verify the positioning accuracy of PPt-YOLOv3 network designed in this paper. YOLOv3 network and tiny-YOLOv3 network are trained as the comparison of PPt-YOLOv3 network on the data set, and the MIOU is tested in the test set. The comparison results are shown in Table 4.

The results show that the MIOU of PPt-YOLOv3 is 5.24% higher than that of YOLOv3 and 8.38% higher than that of tiny-YOLOv3. This shows that on the test

FIGURE 7. The comparison of detection results of images in KITTI dataset with tiny-YOLOv3 and PPt-YOLOv3. Four pictures on the left were detected with tiny-YOLOv3, and four pictures on the right were detected with PPt-YOLOv3. A group of horizontal pictures are the same scene.

TABLE 4. The hardware and software configuration of experimental platform.

Method	MIOU/%
YOLOv3	76.08
tiny-YOLOv3	72.94
PPt-YOLOv3	**81.32**

TABLE 5. The Influence of grid size on detection speed and mAP.

Method	Input	Grid scale	Detection speed /frames·s^{-1}	mAP/%
Tiny-YOLOv3	416×416	13 ×13	188	83.91
	576 ×192	18 ×6	286	68.78
	768 ×384	24 ×12	179	85.41
PPt-YOLOv3	**416 ×416**	**13 ×13**	**183**	**86.21**
	576 ×192	**18 ×6**	**217**	**79.50**
	768 ×384	**24 ×12**	**144**	**91.03**

set, PPt-YOLOv3 produces a higher overlap rate between the prediction box and the original tag box, and has a better accuracy for vehicle positioning. The reason is that through K-means clustering analysis of data sets to select the appropriate size of the candidate box and improve the grid size, we can better improve the positioning accuracy of the model.

In order to reflect the contribution value of each step of improvement to the results, on the basis of the above experiments, several groups of comparative experiments are carried out to analyze the effect of each step of improvement.

In order to study the impact of improving the input size of the network PPt-YOLOv3 on the average accuracy and detection speed, this experiment set up three different sizes of input pictures for test training. As can be seen from Table 5, increasing the number of horizontal grids and changing the grid size can improve the detection accuracy. However, because of the fine mesh division, the number of candidate boxes increases. At the same time, the amount of calculation increases, and the detection speed also decreases.

FIGURE 8. The comparison of detection results of images collected from roads with tiny-YOLOv3 and PPt-YOLOv3. Four pictures on the left were detected with tiny-YOLOv3, and four pictures on the right were detected with PPt-YOLOv3. A group of horizontal pictures are the same scene.

TABLE 6. The Influence of the number of feature maps on detection accuracy.

Method	Number of feature maps	Input image size	Detection speed/ frames·s^{-1}	mAP/ %
Tiny-YOLOv2	1	768×384	204	69.71
Tiny-YOLOv3	2	768×384	179	85.41
Tiny-YOLOv3-3L	3	768×384	170	88.08
PPt-YOLOv3	**3+SPP-net**	768×384	**144**	**91.03**

Table 6 shows the effect of the number of feature maps on the average detection accuracy of the algorithm when the input size is fixed. For a model of 768 × 384 input, tiny-YOLO with 2 and 3 feature maps gets 85.41% and 88.08% mAP, respectively, which is much higher than tiny-YOLOv2 with only 1 feature map (tiny-YOLOv3-3L means that the detection scale is increased to three, but SPP-net is not introduced). Based on the 3-layer feature map, the spatial pyramid pooling module was added, and mAP was increased to 91.03%. At the same time, the increase of feature maps and the addition of the spatial pyramid module have a certain impact on the detection speed of the model.

In order to test the effectiveness of ppt-yolov3 network more intuitively, we select the images in the KITTI datasets and the images collected by the road for detection by two methods, and select eight groups of images in different scenes. Fig. 7 (a) shows the detection results of tiny-YOLOv3 on the test set image, and Fig. 7 (b) shows the detection results of PPt-YOLOv3 on the test set image. For the input picture of the first line, tiny-YOLOv3 recognizes the blocked van incorrectly, while PPt-YOLOv3 recognizes

correctly and positions the three long-distance vehicles in the middle of the picture with higher accuracy; for the picture of the second and third lines, tiny-YOLOv3 repeatedly detects van and tram and misses the blocked car, while PPt-YOLOv3 all detects correctly. In the fourth line, PPt-YOLOv3 recognizes the blocked truck in the distance, tiny-YOLOv3 does not recognize it, which increases the security risk. Through the analysis and comparison of the experimental results, PPt-YOLOv3 detection is better in the test set.

Fig. 8 (a) shows the detection results of tiny-YOLOv3 on the real vehicle road acquisition image, and Fig. 8 (b) shows the detection results of PPt-YOLOv3 on the real vehicle road acquisition image. For the first group of pictures, PPt-YOLOv3 detected and recognized the van in the distance, while tiny-YOLOv3 failed to do so. In the third group of pictures, tiny-YOLOv3 missed car in the picture; in the second and fourth group of pictures, tiny-YOLOv3 failed to detect the blocked car. However, PPt-YOLOv3 was detected correctly. For the four images collected from the road, PPt-YOLOv3 can detect the small target and the occluded target better than tiny-YOLOv3. Based on the above detection results, two kinds of networks have similar detection capabilities for the large-scale non occluded vehicles in the image. For the small-scale vehicles and the occluded vehicles, tiny-YOLOv3 will have missed detection, wrong detection and repeated detection. But the PPt-YOLOv3 proposed in this paper can solve the problem well and detect the vehicle correctly. Therefore, PPt-YOLOv3 has better detection performance.

V. CONCLUSION

Based on the improved tiny-YOLOv3, we propose a new method of vehicle detection, PPt-YOLOv3. Compared with the previous series of YOLO algorithms, PPt-YOLOv3 inducing SPP-net to improve the size of the receptive domain, which has better performance in accuracy, recall, handover and merging ratio and mAP. In particular, it has obvious advantages in mAP, which is 7.12% higher than the original tiny-YOLOv3. The detection speed is reduced by 44 frames/s, but it still far exceeds the real-time requirement. This is because PPt-YOLOv3 increases the detection scale and the number of channels in the feature maps, and changes the input size of the image, which to some extent increases the amount of calculation. Considering the needs of the scene in this paper, in the aspect of vehicle driving in the road environment, it is more important to accurately identify the target. Therefore, it is necessary to sacrifice a small amount of detection speed for the higher detection accuracy. At the same time, because of the simple network structure, the model size of this method is only 51.6MB, which is more convenient for deployment.

PPt-YOLOv3 combines the context features, increases the detection scale, and introduces SPP-net spatial pyramid pooling module in the two detection scales. Increasing the number of feature map channels keeps more target texture features. The new detection scale of 52×52 is more suitable for detecting similar small targets of occluded vehicles, which is very helpful for the detection of distant vehicles. In addition, this method increases the number of transverse grids, refines the grid, and uses K-means clustering method to automatically generate candidate boxes to enhance the characterization ability of the feature map. At the same time, it improves the positioning accuracy of the model, and improves the accuracy of vehicle detection in front of tiny-YOLOv3 network, and has real-time detection speed. PPt-YOLOv3 training is limited by the KITTI data set. Due to the lack of training samples, the detection effect of vehicles in different environments needs to be improved. The future work should be focused on enhancing the generalization ability of the model.

REFERENCES

[1] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[2] G. Richards, "Intelligent cars [control forecasts]," *Eng. Technol.*, vol. 5, no. 1, pp. 40–41, Jan./Feb. 2010.

[3] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "RT3D: Real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3434–3440, Oct. 2018.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[5] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[6] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Mach. Vis. Appl.*, vol. 12, no. 2, pp. 69–83, Aug. 2000.

[7] W. Wang, G. Tian, M. Chen, F. Tao, C. Zhang, A. AI-Ahmari, Z. Li, and Z. Jiang, "Dual-objective program and improved artificial bee colony for the optimization of energy-conscious milling parameters subject to multiple constraints," *J. Cleaner Prod.*, vol. 245, Feb. 2020, Art. no. 118714, doi: 10.1016/j.jclepro.2019.118714.

[8] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 850–864, Mar. 2007.

[9] M. B. van Leeuwen and F. C. A. Groen, "Vehicle detection with a mobile camera: Spotting midrange, distant, and passing cars," *IEEE Robot. Autom. Mag.*, vol. 12, no. 1, pp. 37–43, Mar. 2005.

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: http://arxiv.org/abs/1312.6229

[11] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2421–2424.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, pp. 1–8.

[13] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 32–39.

[14] P. E. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using histogram of oriented gradients features," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 921–928.

[15] Z. Hong, "A preliminary study on artificial neural network," in *Proc. 6th IEEE Joint Int. Inf. Technol. Artif. Intell. Conf.*, Aug. 2011, pp. 336–338.

[16] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M.-R. Akbarzadeh-T, "Vehicle recognition using curvelet transform and SVM," in *Proc. 4th Int. Conf. Inf. Technol. (ITNG)*, Apr. 2007, pp. 516–521.

[17] S. Wu and H. Nagahashi, "Parameterized AdaBoost: Introducing a parameter to speed up the training of real AdaBoost," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 687–691, Jun. 2014.
[18] O. Sharma, "Deep challenges associated with deep learning," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, Feb. 2019, pp. 72–75.
[19] W. W. Zhang, Y. Zheng, Q. Gao, and Z. Mi, "Part-aware region proposal for vehicle detection in high occlusion environment," *IEEE Access*, vol. 7, pp. 100383–100393, 2019.
[20] W. Zhang, Z. Mi, Y. Zheng, Q. Gao, and W. Li, "Road marking segmentation based on siamese attention module and maximum stable external region," *IEEE Access*, vol. 7, pp. 143710–143720, 2019.
[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[22] R. Zhu, X.-J. Mao, Q.-H. Zhu, N. Li, and Y.-B. Yang, "Text detection based on convolutional neural networks with spatial pyramid pooling," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1032–1036.
[23] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[26] B. Yang, Y. Zhang, J. Cao, and L. Zou, "On road vehicle detection using an improved faster RCNN framework with small-size region up-scaling strategy," in *Proc. Pacific-Rim Symp. Image Video Technol.*, Nov. 2017, pp. 241–253.
[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 21–37.
[29] J. Redmon. (Oct. 2017). *Darknet: Open Source Neural Networks in C*. [Online]. Available: http://pjreddie. com/darknet/
[30] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[32] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861
[33] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection," in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 95–101.
[34] K. He, X. Zhang, S. Ren, and 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, Sep. 2015.
[35] M. Tang, Z. Li, and G. Tian, "A Data-Driven-Based wavelet support vector approach for passenger flow forecasting of the metropolitan hub," *IEEE Access*, vol. 7, pp. 7176–7183, 2019.