Road Obstacle Object Detection Based on Improved YOLO V4

Zuo Xiao, Yu Jun

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, Shaanxi, China
E-mail: 1158198898@qq.com

Hu Yuzhe

Jinan University-University of Birmingham Joint Institute
Jinan University
Guangzhou, 511400, Guangdong, China
E-mail: 18137910896@163.com

Xian Tong

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, Shaanxi, China

Hu Zhiyi

Engineering Design Institute
Army Research Loboratory
Beijing, 100000, China
E-mail: 763757335@qq.com

***Abstract*—In recent years, as one of the important technical tasks in the field of deep learning, object detection has broad prospects and applications in the field of road obstacle detection. However, in the real driving scene, there are many obstacles, serious occlusion, overlap and other problems, so that the existing obstacle detection algorithm can not effectively detect the obstacles on the road, so it can not guarantee the driving safety. In order to solve the above problems, this paper improves on the basis of Yolo V4 algorithm. Firstly, kmeans + + clustering is used to generate a priori box suitable for the data set to enhance the scale adaptability; Then, the ciou is used as the loss function of coordinate prediction to evaluate the coincidence degree of prediction frame and truth value frame more reasonably. Finally, a suitable target detection data set is constructed by preprocessing the public data set cityccaps. The experimental results show that the improved algorithm can achieve more than 90% accuracy for obstacles with large number of targets in the training set. Compared with the original Yolo V4, the average detection accuracy of the improved algorithm is improved by 2.03%.**

Keywords-YOLO v4 Algorithm; Obstacle; Object Detection; Loss Function

I. INTRODUCTION

As the main means of transportation for travel, cars provide great convenience to our lives, but the problem of safe driving of cars on the road comes with it. According to the National Statistical Yearbook, a total of 247,646 road traffic accidents occurred nationwide in 2019, causing more than 310,000 casualties and property losses of 1,346.1 million yuan. In order to reduce the occurrence of such problems and improve driving safety, object detection technology has been gradually applied to the field of car assisted driving [1]. It can provide vehicles with the perception information of the surrounding environment and automatically detect road obstacles to improve the road. The purpose of driving safety.

In recent years, scholars at home and abroad have gradually applied deep learning technology to obstacle object detection [2]. Prabhakar [3] and others have developed a set of deep learning systems on assisted driving for the detection and classification of road obstacles such as vehicles, pedestrians, animals, etc., suitable for autonomous driving [4] cars driving on highways. Tang Bowen [5] and others used the YOLO v3 algorithm to

DOI: 10.21307/ijanmc-2021-023

complete the UAV obstacle detection. The speed is fast but the accuracy of identifying the position of the object is poor. Guo Jishun [6] and others introduced the dynamic residual network to solve the problem of deep network and poor generalization in object detection, which solved the degradation of deep neural network well, but did not completely solve the performance problem caused by network deepening. The detection speed and accuracy of the above methods need to be improved. In this paper, YOLO v4 [7] of the YOLO series is used for object detection. Compared with YOLO v3, this algorithm lowers the training threshold and uses a single GPU for training more effective. More importantly, it has a significant improvement in detection speed.

Although the yolo v4 algorithm is considerable in terms of accuracy and detection speed, it still has some shortcomings, such as the random initial clustering center of the anchor box a priori box generated by the Kmeans method, resulting in inaccurate clustering results; The too high coincidence of urban environment makes it difficult to predict coordinates, which leads to the low accuracy of the detection results.

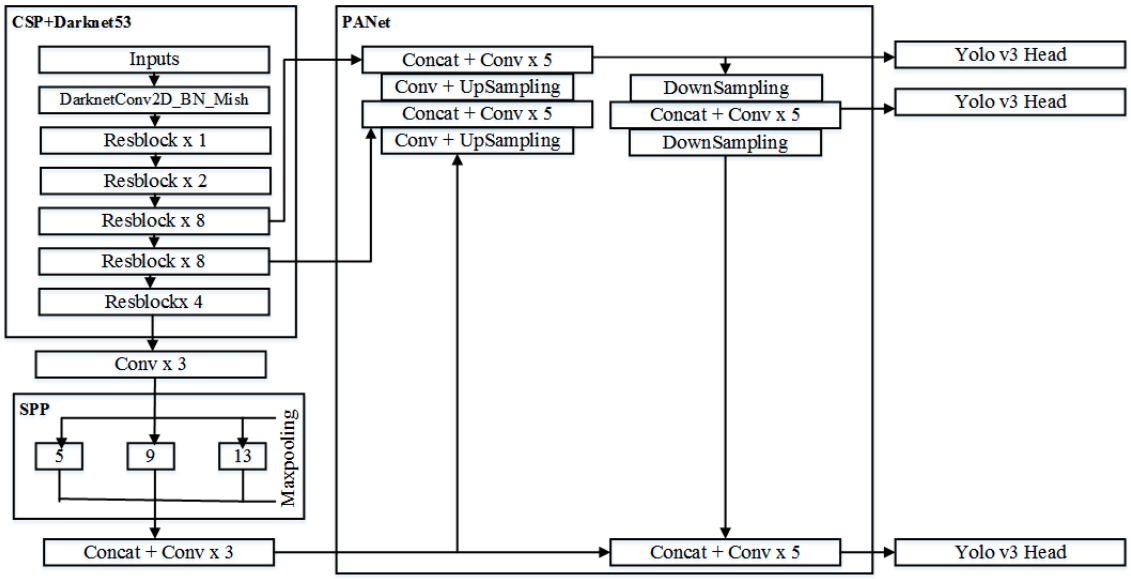


Figure 1. Network structure of YOLO v4 algorithm

TABLE I. ENVIRONMENT CONFIGURATION

Hardware environment	**processor**	**Intel(R) XEON W-2133**
	Graphics card	**Nvidia TITAN XP 12G**
Software Environment	operating system	Ubuntu 16.04
	Deep learning framework	Tensorflow-gpu
	Programming language	Python
	translater	Pycharm2019.1

In order to solve such problems, this paper proposes an improved YOLO V4 object detection algorithm. By improving the network structure of the algorithm, improving the Kmeans clustering and optimizing the coordinate prediction loss function, the improved algorithm is more suitable for object detection of road obstacles.

II. YOLO V4 ALGORITHM PRINCIPLE

YOLO (YouOnlyLookOnce) [8] network is a kind of object detection algorithm based on regression. Its main idea is to divide the image into multiple grids, then use the depth neural network to judge whether the network has a object or not, and then predict the category and position of the object. The network structure of YOLO v4 is shown in Figure 1. Keeping the Head part of YOLO v3, the CSPDarknet53 module selected by the backbone network, introduces spatial pyramid pooling (SPP) as an additional module of the Neck part to expand the receptive field, and PANet's path aggregation module is used as a part of the Neck. Among them, Darknet53 contains 5 residual blocks, and the number of small residual units contained in the residual blocks are 1, 2, 8, 8, and 4 respectively. CSPDarknet53 modifies Darknet53. Each large residual block is added with a CSPNet module and integrated into the feature map through gradient descent. Part of the feature map is convolved, and the other part is combined with the previous convolution result. CSP can improve the ability of convolutional neural networks to extract features and improve computational efficiency. PANet (Path Aggregation Network) makes full use of feature fusion. YOLO v4 also changes the fusion method from addition to multiplication, so that the network can get accurate detection results. YOLO v4 introduces Mosaic data augmentation and SAT for data enhancement, genetic algorithm selects hyperparameters, uses cross-small batch normalization, and uses DropBlock [9] regularization. They lowered the training threshold, allowing the model to get fast and accurate detection results under ordinary GPU conditions.

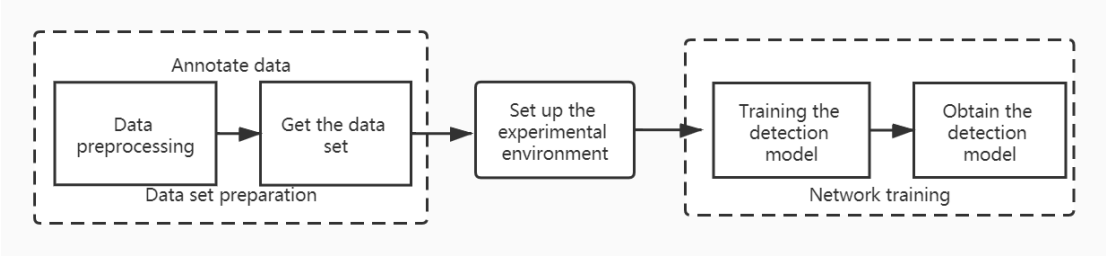


Figure 2. Experimental framework

Although the YOLO v4 algorithm has excellent accuracy and detection speed, there are two problems:

1) The anchor box value (anchor box is a priori box) generated by the Kmeans method, random initial aggregation,the clustering result is not necessarily accurate due to the cluster center, thus affecting the accuracy of the detection result.

2) When the coincidence degree of the object is relatively high, a good coordinate prediction loss function is required to obtain the actual position of the object.

III. IMPROVED YOLO V4 ALGORITHM DESIGN

In response to the above problems, this article has improved the YOLO v4 algorithm. The main work includes:

Figure 3. Annotation of car instance segmentation

```
"label": "car",
"polygon": [[1357, 725],  [1339, 654], [1320, 600],  [1297, 544],  [1279, 510], [1282, 507],
            [1286, 499],  [1284, 489], [1277, 480],  [1266, 477],  [1217, 387], [1209, 376],
            [1182, 351],  [1166, 343], [966, 329],  [941, 330],  [937, 328],  [937, 322],
            [936, 318],  [931, 316],  [925, 321],  [925, 326], [924, 328],  [716, 338],
            [690, 344], [636, 396],  [619, 427],[585, 485], [559, 529], [532, 572],[496, 687],
            [490, 733], [485,  782], [487, 819], [489, 843],  [491, 880], [494, 898],[496, 926],
            [501, 949], [506, 964], [513, 976], [528, 981], [555, 993], [590, 993], [1344, 905],
            [1345, 866], [1354, 818],[1358, 775],  [1354, 760], [1354, 743], [1356, 730]]
```


Figure 4. The json file of the car label


```
train/bremen/bremen_000010_000019_leftImg8bit.png 1713,389,1758,504,0 1672,386,1712,502,0 1802,425,2020,505,1
```


Figure 5. The txt file of the image tag

1) Kmeans++ is selected for the generation of anchor box;

2) The coordinate prediction loss function uses CIoU.

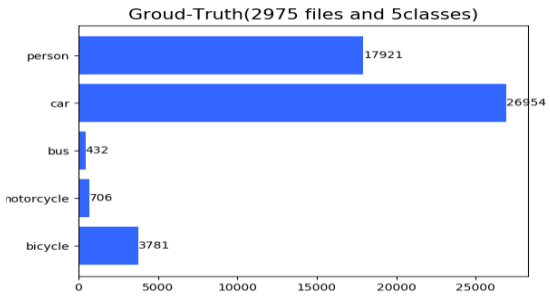
A. Generate anchor box with Kmeans++

The YOLO v4 algorithm originally used the Kmeans clustering algorithm to generate the anchor box. Since the initial clustering center of the Kmeans algorithm is randomly selected, the classification results may not be accurate. The selection of the clustering center must be as far away as possible. Therefore, this paper uses the Kmeans++ clustering algorithm to analyze the data set and generate suitable anchor box values. The Kmeans++ algorithm ensures that the latest cluster center is as far away as possible from the previous center. In order to reduce the error caused by the size of the anchor box itself, Intersection over Union (IoU) is selected as the measurement standard, and the calculation formula is shown in formula (1). Among them, box is the object truth box, centroid is the obtained a priori box, and IoU (box, centroid) represents the intersection ratio of the a priori box and the truth box. It can be seen that the smaller the distance d, the larger the intersection ratio, the more the a priori box and the truth box overlap, and the better the clustering effect.

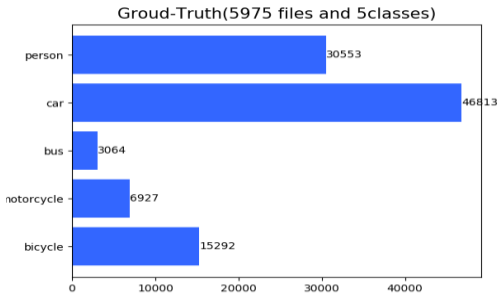
$$d(box, centroid) = 1 - IoU(box, centroid) \quad (1)$$

Figure 6. Add Gaussian noise

Figure 7. Median fuzzy processing



(a) Number of objects before amplification



(b) Number of objects after amplification

Figure 8. Object number before and after data amplification

TABLE II. MAIN NETWORK PARAMETER VALUES

Parameter	Value	Parameter	Value
LEARN_RATE_INIT	1e-4	MOVING_AVE_DECAY	0.9999
LEARN_RATE_END	1e-6	STAGE_EPOCHS	100

Based on the three output scales, three types of anchor boxes are set, and 9 types of anchor boxes are clustered in this paper. The anchor box values are (54, 56), (93, 89), (207, 161), (60, 109), (133, 125), (145, 257), (85, 167), (254, 188) and (293, 286).

B. *Choose CIoU as the loss function of coordinate prediction*

In object detection, the method for the model to evaluate the distance between the predicted frame and the true value frame usually adopts IoU, GIoU and DIoU. However, there are the following problems: IoU is a ratio, which is not sensitive to the size of the object, and cannot directly optimize the non-coincident range; GIoU can detect the non-coincident range but does not consider the center distance; DIOU considers the bounding box coincidence and center distance problems but does not Consider the scale ratio. In response to the above problems, this paper uses Complete Intersection over Union (CIoU) as the coordinate loss function, which takes into account the overlap area, center distance and scale ratio, so it can more reasonably evaluate the degree of overlap between the prediction box and the true value box.

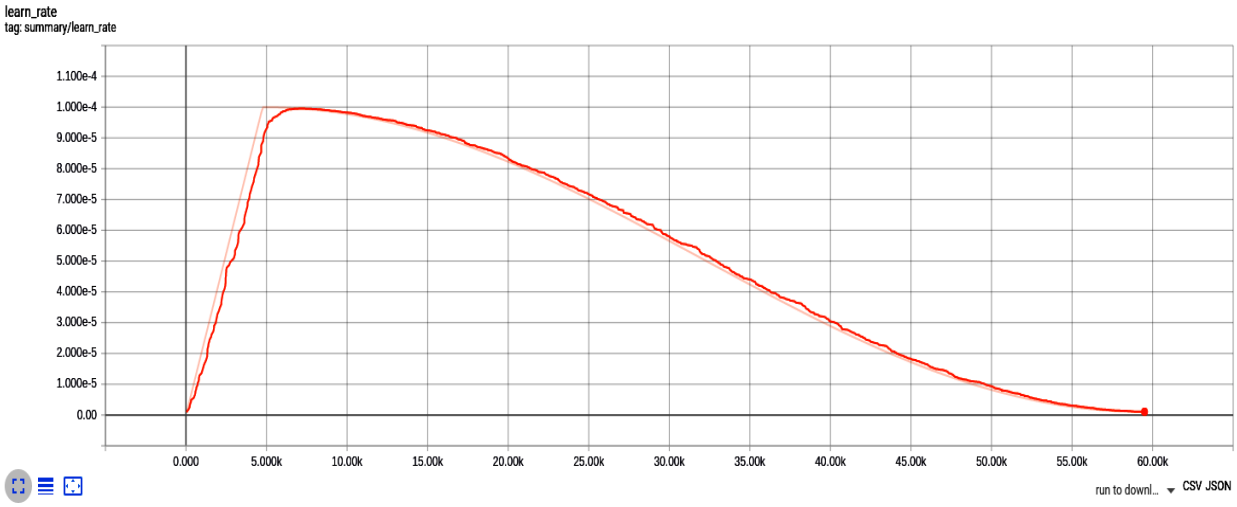


Figure 9. Learning rate change curve

CIoU adds an influence factor on the basis of the penalty item of DIoU, and considers the fitting degree of the aspect ratio of the predicted frame to the aspect ratio of the real frame as the consideration range, as shown in formula (2).Where v is a parameter used to measure the consistency of the aspect ratio， $\alpha$is the trade-off parameter, b represents the center of the prediction box, represents the center of the real box, $p^2()$ represents the square of the Euclidean distance, and c represents the minimum diagonal distance between the prediction box and the real box in the bounding box. The specific calculation methods of $v$and $\alpha$are shown in formulas (2) and (4), and the loss function of CIoU is shown in formula (5).

CIoU has scale invariance. When the object frame overlaps and contains, the normalized distance between the predicted frame and the real frame is minimized, thereby speeding up the convergence speed, making the regression process more stable, and avoiding divergence problems during the training process.

$$CIoU = IoU - \frac{p^2(b, b^{qt})}{c^2} - \alpha v \tag{2}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^g}{h^{8t}} - \arctan\frac{w}{h})^2 \tag{3}$$

$$\alpha = \frac{v}{(1-IoU)+v} \tag{4}$$

$$L_{CloU} = 1 - CIoU \quad (5)$$

IV. EXPERIMENT AND RESULT ANALYSIS

In this article the experiment is carried out under the Linux system, and the

The experiment in this article is carried out under the Linux system, and the experiment environment is shown in Table 1. In order to reduce the training time of the deep neural network model and increase the calculation speed, the Nvidia TITAN XP12G graphics card is used, and CUDA9.0 and cuDNN7.0 are configured to call the GPU for acceleration. The deep learning framework chosen is Tensorflow.

The overall framework of the experiment is shown in Figure 2. It mainly includes three parts: the preparation of the data set, the construction of the training environment and the training of the network. The strategy of network training is as follows

A. Preparation of experimental data set

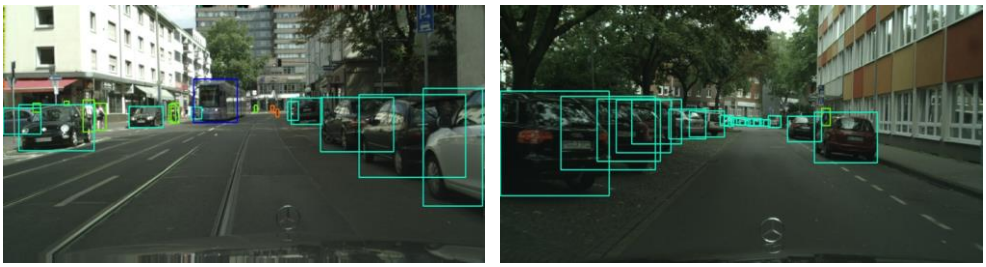
The data in this paper comes from Cityscapes, which mainly contains 5,000 high-quality pixel-level annotated images of driving scenes in urban environments and 20,000 rough-annotated images. Since the official does not provide annotations for the test set images, we used a training set of 2975 images for training and a verification set of 500 images for testing.

We mainly adopt two methods: data annotation and data amplification.

1) Data labeling

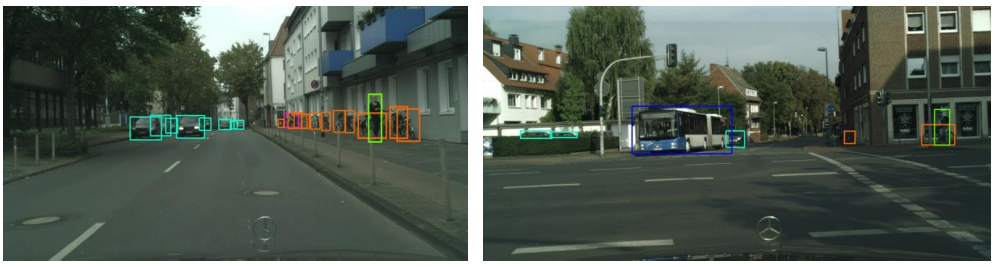
The Cityscapes dataset provides annotation information for semantic segmentation and instance segmentation. Generating a json file to store the outline information of the sample in the labeling method shown in Figure 3. As shown in Figure 4, the json file stores all the coordinate information of the contour points. The txt file shown in Figure 5 saves the coordinate information and target category information of all objects in the image.

2) Data amplification



(a) Images with a large number of targets (b) The image of the target being occluded

Figure 10. Experimental results based on Yolo v4



(a) Images with a large number of objects (b) The image of the objects being occluded

Figure 11. Experimental results based on improved YOLO v4

Data amplification adopts two methods: adding noise and blurring. Figure 6 shows the effect of adding Gaussian noise to the original image. Figure 7 shows the effect of median blur on the original image in the Cityscapes training set, we selected 500 images containing buses, bicycles, and motorcycles for data augmentation. The number of objects in each category before amplification is shown in Figure 8(a), and the number of objects in each category after amplification is shown in Figure 8(b).

B. Model training

In this paper, the detection model is trained according to the parameters shown in Table 2. Among them, BLEARN_RATE_INIT represents the minimum learning rate; LEARN_RATE_END represents the maximum learning rate; MOVING_AVE_DECAY represents the moving average, which is used to estimate the local mean value of some parameters, so that the parameter update is related to the historical value within a period of time. STAGE_EPOCHS represents the iterative training of the data set 100 Table 3 Comparison of test results times. In the actual training process, each training batch will save one model, and it is set to save up to 10 models.

TABLE III. COMPARISON OF TEST RESULTS

AP(%)	Car	Bus	Person	Motorbike	Bicycle	mAP(%)
yolo v4	0.98	0.93	0.92	0.81	0.51	82.95
Improved yolo v4	0.99	0.93	0.92	0.81	0.58	84.98

The strategy of network training during the experiment is as follows.

1) Multi-scale training strategy.

When the detection network is trained, the multi-scale training rules can be used to input different image resolutions, and the model can detect the input small-scale images faster. The specific training method is to modify the input size of the image every few batches so that the model is robust to images of different sizes and can detect images of different scales.

2) Warm up strategy

The data set of the early network is generally small and the network is not deep, so there is no need to adopt the Warm up strategy. As computer vision tasks become more and more complex, setting a fixed learning rate will cause problems in the training process. The best learning rate setting method is to change the learning rate according to the iterative batch, which not only has good training efficiency, but also avoids the instability of the fully connected layer, thereby increasing the deep similarity of the model. During the training process, the learning rate changes with the training batches are shown in Figure 9. The horizontal axis represents the iteration batches, and the vertical axis represents the learning rate. The curve in the figure reaches the highest learning rate at 5000 iterations. The learning rate at different stages is different. First, use a larger learning rate to find the global optimum, and then use a smaller learning rate to find the local optimum to obtain the global optimum solution of the model.

C. Experimental results and analysis

This paper uses the improved YOLO v4 network model before and after the experiment respectively, and the results of the visualization experiment based on YOLO v4 are shown in Figure 10. The visual experiment results based on the improved YOLO v4 are shown in Figure 11. The picture contains urban street scenes under different light levels, including five types of obstacles such as cars, buses, and motorcycles. They are marked by five different color detection boxes. These obstacles are overlapped, exposed to varying degrees of distance and distance.

It can be seen from Figure 10 that most of the obstacles included in the images with many objects and the images with heavy object occlusion have been detected, including the car roof can also be accurately identified. However, there are some problems such as too small positioning box, repeated box and missing detection. The obvious problem is: when people ride motorcycles or bicycles, most of the cases will not detect people, and the detection frame of

motorcycles or bicycles is too large, in addition, the detection frame of large buses can not detect the whole car body. It can be seen from Figure 11 that various obstacles can be detected more effectively in the same background. Under different backgrounds, different forms of obstacles can be effectively detected, including obstacles blocked by foreign objects, overlapping objects, incomplete shooting, and blurred pixels. Riders on bicycles and motorcycles can be effectively detected, there is no redundant detection frame, and the coordinate information of obstacles can be predicted more accurately.

The test results of the above-mentioned comparative test are shown in Table 3. It can be seen that the average accuracy of the improved YOLO v4 algorithm is 2.03% higher than the detection result before the improvement.

V. Conclusion

This paper proposes a object detection algorithm based on improved YOLO v4, trains and tests the objects detection network on the Cityscapes dataset. Experimental results show that the improved YOLO v4 algorithm in this paper improves the average recognition accuracy of vehicle objects, and solves the problem of poor accuracy caused by different initial clustering centers in the YOLO v4 algorithm due to different clustering results and the lack of optimized bounding boxes Problem. Compared with YOLO v4, the detection accuracy of the improved algorithm is increased by 2.03%.

References

[1] Zhao Richeng. Research on road obstacle detection technology in assisted driving [D]. Xidian University, 2015.

[2] Wang Tiantao, Zhao Yongguo, Chang Faliang. Obstacle detection based on visual sensor [J]. Computer Engineering and Applications, 2015, 51(4):180-183.

[3] Prabhakar G, Kailath B, Natarajan S, et al. Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving[C]//2017 IEEE region 10 symposium (TENSYMP). IEEE, 2017:1-6.

[4] Zeng Weiliang, Wu Miaosen, Sun Weijun, et al. Overview of Research on Autonomous Taxi Dispatching System [J]. Computer Science, 2020, 47(05):189-197.

[5] Tang Bowen. Research on Obstacle Detection and Obstacle Avoidance Processing During UAV Driving [D]. Guangxi University of Science and Technology, 2019.

[6] Guo Jishun. Semantic segmentation and target detection technology for autonomous driving [D]. University of Electronic Science and Technology of China, 2018.

[7] Zhang Xin, Qi Hua. Research on human abnormal behavior detection algorithm based on yolov4 [J]. Computer and digital engineering, 2021,49 (04): 791-796.

[8] Wong A , Famuori M , Shafi Ee M J , et al. YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection [J]. 2019.

[9] Wang J, Gao F, Dong J, et al. Adaptive DropBlock-Enhanced Generative Adversarial Networks for Hyperspectral Image Classification [J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, PP(99):1-14.