YOLOv3 tiny vehicle: A new model for real-time vehicle detection

Xiuxin Ma[a], Huaiyu Zhuang[b], Jiaxian Deng[a], Jia Ren[a,c], Yani Cui[a*]
[a] School of Information and Communication Engineering, Hainan University, Haikou, Hainan, China; [b] China Mobile Financial Technology Co., Ltd, Beijing, China; [c] The Institute of Naval Industrial Design, Shandong University of Art & Design, Qingdao, Shandong, China

ABSTRACT

Existing vehicle detection has the problem of unbalanced detection accuracy and speed. Aiming at this problem, this paper proposes a new real-time vehicle detection model named YOLOv3 Tiny Vehicle. The proposed network replaces the Maxpooling layers of the original network with the convolutional layers to ensure that the characteristic information of the vehicle was preserved to the greatest extent. On this basis, our work adds a dense connection structure to the original network, which greatly reduces or even eliminates the overfitting problem during network training. The experimental results show that the mean Average Precision (mAP) of the model on the Beijing Institute of Technology vehicle (BIT-Vehicle) dataset can reach 96.80%, the Frames per second (FPS) can reach 188. At the same time, it also shows that our model has preeminent generalization ability.

Keywords: YOLOv3 tiny, vehicle detection, dense module, deep convolutional neural network

1. INTRODUCTION

Vehicle detection usually extracts vehicle information from pictures or video sequences containing vehicles. It is one of the typical examples of object detection applied in intelligent transportation systems. It can be used for vehicle identification, detection, tracking, capturing vehicle violations, and controlling traffic. Therefore, researchers have proposed a great many vehicle detection methods. The existing vehicle detection methods can be divided into traditional methods and deep convolutional neural network-based methods.

Traditional vehicle detection algorithms generally combine machine learning classifiers with feature extractors. Wei et al.[1] used the Histogram of Oriented Gradients (HOG) for feature extraction and, on this basis, used Support Vector Machine (SVM) for classification to achieve vehicle detection and tracking in complex urban backgrounds. A vehicle detection method based on HOG and Local Binary Pattern feature fusion (LBP) [2] was proposed. This method utilizes HOG-LBP fusion features to train vehicle classifiers and improve vehicle detection accuracy in terrible weather conditions. The methods based on feature extraction can perform well in vehicle detection. However, this method has complicated steps, low detection accuracy, and poor generalization ability.

The object detection algorithms based on the deep convolutional neural network are mainly divided into two-stage and one-stage target detection algorithms. In the first step, the two-stage algorithm usually presents the candidate regions, and the second step conducts the subsequent classification and positioning based on the candidate regions. Typical two-stage detection algorithms include R-CNN[3], Fast R-CNN[4], and Faster R-CNN[5]. Unfortunately, the detection performance of these methods will greatly decrease when the size of the vehicle changes dramatically due to rapid vehicle movement. The second-stage detection algorithm performs well in accuracy overall, but its time cost is vast.

To enhance the detection speed, You Only Look Once (YOLO)[6], a one-stage target detection algorithm that is emblematic, was proposed by Redmon et al. YOLOv1[6] is very fast but is less accurate than Faster R-CNN, especially for small targets. Due to the poor detection effect of YOLOv1 on small targets, Redmon et al. investigated YOLOv2[7] and mainly made some improvements on YOLOv1, which can balance the speed and accuracy well and accurate detection of precise small targets. To further improve object detection accuracy, Redmon et al. proposed YOLOv3[8], which significantly improved the detection accuracy while maintaining the speed. Due to the apparent superiority of this algorithm, the majority of scholars have applied YOLOv3 to vehicle detection. Xu et al.[9] increased the depth of YOLOv3 and used the network for vehicle detection to enhance the detection accuracy at the expense of the detection speed of the

* cyn0213@163.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125061Z
 · 0277-786X · doi: 10.1117/12.2661789

network. Luo et al.[10] presented the Dense YOLOv3 algorithm, reducing the overfitting phenomenon and achieving good detection results.

Tiny YOLO is one of the most widely used algorithms in the YOLO family, with fewer parameters and a faster learning speed. A real-time model for vehicle detection, Little YOLO-SPP, based on YOLOv3 tiny was proposed[11]. This model can detect vehicles more accurately, but the detection types are limited, only divided into buses and cars. A YOLOv3 tiny-based vehicle detection method was investigated for reaching 17 FPS on a general configuration computer with an accuracy of 95.05%[12]. Recently, Bochkovskiy et al. proposed YOLOv4[13] networks, improving their detection accuracy and processing speed. However, the YOLOv4 networks showed poor detection performance against nighttime scenarios. An improved YOLOv4 vehicle detection network was proposed[14], but the detection effect has yet to be improved. Soon after YOLOv4, Jocher presented the YOLOv5[15], which improved detection speed. Wu et al.[16] introduced the ghost module based on YOLOv5, which significantly improved the vehicle detection speed, but the detection accuracy decreased.

An improved YOLOv3 tiny network came up with seemly set the problems of detection accuracy, unbalanced detection speed, and poor model generalization ability of the current algorithm. The mAP of our method reaches 96.80%, and the FPS goes 188. Compared to Faster R-CNN+ResNet[17], our mAP improves by 5.52% and is faster.

The remainder of the article is roughly organized as follows: The basic theoretical method of the article and improved YOLOv3 tiny network are presented in Section 2. Experiments were performed as shown in Section 3. For Conclusions, see Section 4.

2. METHODOLOGY

2.1 YOLOv3 tiny

YOLOv3 tiny is a plain version derived from YOLOv3, with fewer parameters and running faster than YOLOv3, but its detection accuracy is slightly worse than YOLOv3. The dataset used in the paper was collected from high-speed intersections, with little background change, and did not require much processing, so it is more suitable for the YOLOv3 tiny algorithm.

YOLOv3 tiny follows the grid idea of the YOLO series. The input image is divided into S×S grids by the network. If the center of an object falls within a grid interval, the task of predicting this object falls on the grid. Each grid is responsible for predicting B bounding boxes, and each bounding box is responsible for predicting four position parameters x, y, w, h, and one confidence degree. In addition to this, each grid also indicates scores for C categories. From this, the calculation formula of the network is obtained:

$$\mathrm{S}\times S\times\left(B\times 5+C\right) \tag{1}$$

where (x, y) is the central coordinate of the predicted bounding box, which is relative to the grid, the width of the predicted bounding box is denoted by w, the height of the predicted bounding box is denoted by h, and w and h are relative to the entire image. Confidence means the accuracy of the positioning, that is, the intersection ratio between the predicted bounding box and the truth bounding box. The formula for the confidence is as follows:

$$Confidence=P_r(object)\times IOU_{pred}^{truth} \tag{2}$$

In equation (2), $P_r(object)$ represents the probability of detecting an object, 1 if an object is detected, and 0 otherwise.

2.2 K-means++ Clustering

The original network YOLOv3 tiny anchors are primarily tall and thin bounding boxes. At the same time, the image size of the BIT-Vehicle dataset is too large, and most of the bounding boxes are close to square or wide and high; the original anchor is not fit for the BIT-Vehicle dataset. The method of this paper of K-means++ clustering redefines a new set of anchor boxes on the BIT-Vehicle dataset to predict the coordinates of the bounding boxes, and the size of the new anchor box is more suitable for vehicle detection. In equation (3), the intersection ratio between the boundary box and the corresponding anchor represents the distance between samples.

$$d\left(bbox,centroid\right)=1-IOU\left(bbox,centroid\right) \tag{3}$$

Since there were six different vehicle types in the dataset, the K was set to 6. The final calculated anchor boxes sizes are [81, 98], [135, 128], [147, 144], [157, 165], [168, 197], and [233, 191]. The shape of the new anchor box is more suitable for the shape of the vehicle, making the network training faster and better, and improving the positioning accuracy.

2.3 YOLOv3 tiny vehicle

Inspired by the YOLOv3 tiny network, this part, of this paper optimizes it. The improved YOLOv3 tiny network addresses two defects, adequately, of the primordial network: 1) Misclassification of confusing vehicles is easy. Due to the different camera positions, angles and distances, the network can easily confuse different types of vehicles, such as SUV and Sedan, during detection. 2) The original network has a simple structure and cannot fully use all feature information. Based on the YOLOv3 tiny network, the paper primary put forward the improvement approach:

(1) Convolutional layers replace all Maxpooling layers in the original network with stride 2 and size 3×3. At the cost of slightly increasing network parameters, the improved network effectively reduces inter-class confusion, and the detector trains better.

(2) A dense module is added to the backbone to fully extract features without increasing the complexity of the network, effectively reducing overfitting and improving detection accuracy.

(3) YOLOv3 Tiny Vehicle contains 22 convolutional layers deeper than the original YOLOv3 tiny network. Still, only about 7.28M parameters need to be trained due to the dense module, the network complexity is lower than the YOLOv3 tiny.

The improved network structure, as a whole, is made up of a Backbone and Head, which is shown in Figure 1. Our network has 22 convolutional layers. Backbone is used to extract features. Backbone is used to extract features. It is composed of a convolutional layer and 5 Dense blocks. Except for the number of convolution kernels, each Dense block's size of the blocks' convolutional layers is all the same. The Head comprises different convolutional layers for predicting object categories and bounding boxes. Table 1 shows the specific improved network model.

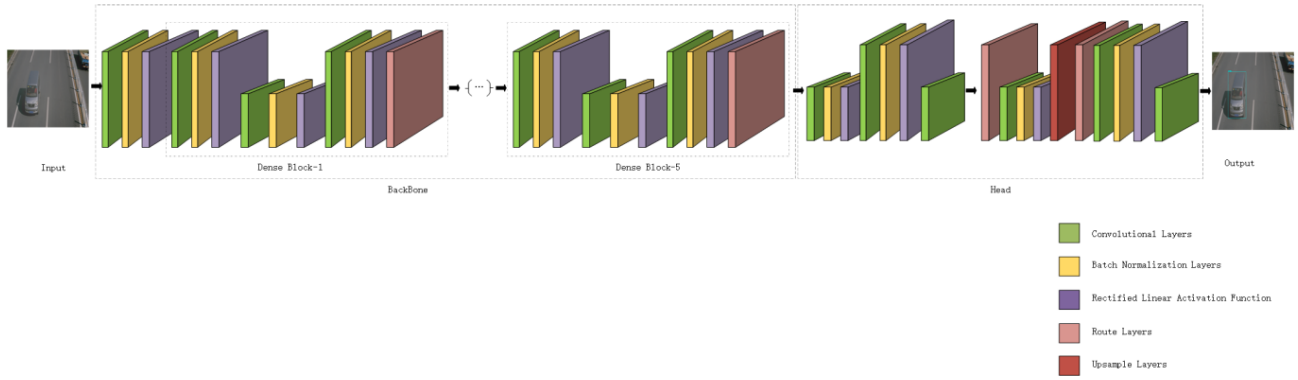

Figure 1. YOLOv3 Tiny Vehicle network structure diagram.

3. EXPERIMENTS

3.1 Experimental environment

The experiments in this paper were performed using the PyTorch deep learning framework on an Ubuntu 18.04.6 64-bit PC equipped with an Intel ® Xeon (R) CPU E5-2680 v3 @ 2.50GHz 48 and Nvidia GeForce GTX TITAN X GPU. The specific experimental environment is shown in Table 2.

3.2 Performance evaluation indicators

This paper uses the following three evaluation metrics to assess and compare the proposed models, namely average precision (AP), mAP, and FPS. AP defines the average accuracy of a single class, and its calculation formula is:

$$AP_i = \int_0^1 p_i(R_i) dR_i \tag{4}$$

Table 1. The difference between the YOLOv3 tiny network structure with the proposed network.

YOLOv3 tiny				YOLOv3 tiny vehicle			
Type	Filters	Size/stride	Output	Type	Filters	Size/stride	Output
Input			416*416*3	Input			416*416*3
Conv-1	16	3*3	416*416*16	Conv-1	16	3*3	416*416*16
Maxpool-1		2*2/2	208*208*16	Conv-2	32	3*3/2	208*208*32
Conv-2	32	3*3	208*208*32	Conv-3	16	1*1	208*208*16
Maxpool-2		2*2/2	104*104*32	Conv-4	32	3*3	208*208*32
Conv-3	64	3*3	104*104*64	Route	-1,-3		208*208*64
Maxpool-3		2*2/2	52*52*64	Conv-5	64	3*3/2	104*104*64
Conv-4	128	3*3	52*52*128	Conv-6	32	1*1	104*104*32
Maxpool-4		2*2/2	26*26*128	Conv-7	64	3*3	104*104*64
Conv-5	256	3*3	26*26*256	Route	-1,-3		104*104*128
Maxpool-5		2*2/2	13*13*256	Conv-8	128	3*3/2	52*52*128
Conv-6	512	3*3	13*13*512	Conv-9	64	1*1	52*52*64
Maxpool-6		2*2/1	13*13*512	Conv-10	128	3*3	52*52*128
Conv-7	1024	3*3	13*13*1024	Route	-1,-3		52*52*256
Conv-8	256	1*1	13*13*256	Conv-11	256	3*3/2	26*26*256
Conv-9	512	3*3	13*13*512	Conv-12	128	1*1	26*26*128
Conv-10	33	1*1	13*13*33	Conv-13	256	3*3	26*26*256
YOLO-1	Mask=3,4,5			Route	-1,-3		26*26*512
Route	-4		13*13*256	Conv-14	512	3*3/2	13*13*512
Conv-11	128	1*1	13*13*128	Conv-15	256	1*1	13*13*256
Upsample	Stride=2		26*26*128	Conv-16	512	3*3	13*13*512
Route	-1,8		26*26*384	Route	-1,-3		13*13*1024
Conv-12	256	3*3	26*26*256	Conv-17	256	1*1	13*13*256
Conv-13	33	1*1	26*26*33	Conv-18	512	3*3	13*13*512
YOLO-2	Mask=1,2,3			Conv-19	33	1*1	13*13*33
				YOLO-1	Mask=3,4,5		
				Route	-4		13*13*256
				Conv-20	128	1*1	13*13*128
				Upsample	Stride=2		26*26*128
				Route	-1,13		26*26*384
				Conv-21	256	3*3	26*26*256
				Conv-22	33	1*1	26*26*33
				YOLO-2	Mask=1,2,3		

Table 2. Hardware and software experimental environment.

Name	Version
OS	Ubuntu 18.04.6 64bit
CPU	Intel® Xeon(R) E5-2680 v3 @ 2.50GHz × 48
GPU	Nvidia GeForce GTX TITAN X GPU
CUDA	10.1.105
cuDNN	7.6.4
PyTorch	1.6.0
PyThon	3.7

where P_i is the precision for a single class, and R_i is the recall for a single class. The mAP is one of the essential evaluation metrics in multi-classification tasks, and its definition is as follows:

$$mAP = \frac{1}{n}\sum_{i}^{n} AP_i \tag{5}$$

Where the AP_i is the average accuracy of a single class and the number of data categories is n, and FPS, a vital evaluation indicator for real-time detection tasks, is the number of images the model can be conducted per second.

3.3 Results and discussion

3.3.1 Analysis of the Experimental Results. This paper adopts two datasets named Beijing Institute of Technology (BIT)-Vehicle and CompCars[18] which are collected from road monitoring. To further investigate the generalization ability of the proposed algorithm, we randomly selected 1200 images on the CompCars dataset and manually annotated them as another test set to make it more comprehensive and objective than other model comparisons.

As shown in Table 3, the proposed network has made three main adjustments. The first adjustment (Step 1) is to replace the Maxpooling layer in the original network with the convolutional layer. This improvement increases mAP by about 1.4% compared to the original YOLOv3 tiny. The second adjustment (Step 2) is to add a dense module to the network, which significantly improves the detection accuracy, with mAP reaching 95.60%, which also shows that this improvement measure is effective. The third improvement measure (Step 3) is to retrain the anchor boxes that are more suitable for vehicles on the BIT-Vehicle dataset, which helps the network to train faster and better, and the mAP of the network is also improved by about 1.2%. Overall, the proposed network is less complex than the original YOLOv3 tiny, and the detection effect improves by 3.5% compared to the mAP of the YOLOv3 tiny, and the detection speed reaches 188 FPS, which is about 5 milliseconds of inference time. The frame rate of our algorithm can effortlessly satisfy the requirements of real-time vehicle detection.

Table 3. mAP, AP, FPS and complexity of different models.

Model	mAP (%)	Bus (%)	Microbus (%)	Minivan (%)	Sedan (%)	SUV (%)	Truck (%)	FPS (%)	No. of Parameters (M)
YOLOv3 tiny	92.50	100	90.77	93.33	98.13	90.03	82.72	142	8.68
Step 1	93.90	100	87.86	92.90	97.25	92.52	92.60	212	11.83
Step 2	95.60	99.71	90.32	93.28	97.41	96.53	96.43	188	7.28
Step 3 (YOLOv3 tiny vehicle)	96.80	100	94.05	96.08	97.55	95.85	97.35	188	7.28

To evaluate the proposed model more objectively, Table 4 compares the recently proposed and excellent vehicle detection algorithm with the proposed algorithms. The detection performance of the proposed model is almost better than all the models in Table 4, except for YOLOv3, although the detection accuracy of the proposed model is slightly inferior to that of YOLOv3, its inference speed is better than the latter.

Table 4. Performance comparison of various models.

Model	mAP (%)	Bus (%)	Microbus (%)	Minivan (%)	Sedan (%)	SUV (%)	Truck (%)	Time(s) per image
Faster R-CNN +ResNet[17]	91.28	90.62	94.42	90.67	90.63	91.25	90.07	0.680
YOLOv3[12]	96.88	98.99	96.54	92.17	96.90	99.99	96.69	0.09
Trimmed Tiny-YOLOv3[12]	95.05	100	95.38	91.88	99.12	89.59	94.30	0.019
Step3(YOLOv3 Tiny Vehicle)	96.80	100	94.05	96.08	97.55	95.85	97.35	0.005

The performance of our algorithm was presented in Figure 2. Figures 2a-2c contain multiple vehicles, indicating that the proposed network performs well when detecting various targets. The mAP of Sedan and Microbus in Figure 2a reach 95% and 94%, respectively. Figures 2d-2i contain six vehicle types in the dataset, and the results indicate that the proposed algorithm can accurately detect the vehicle.

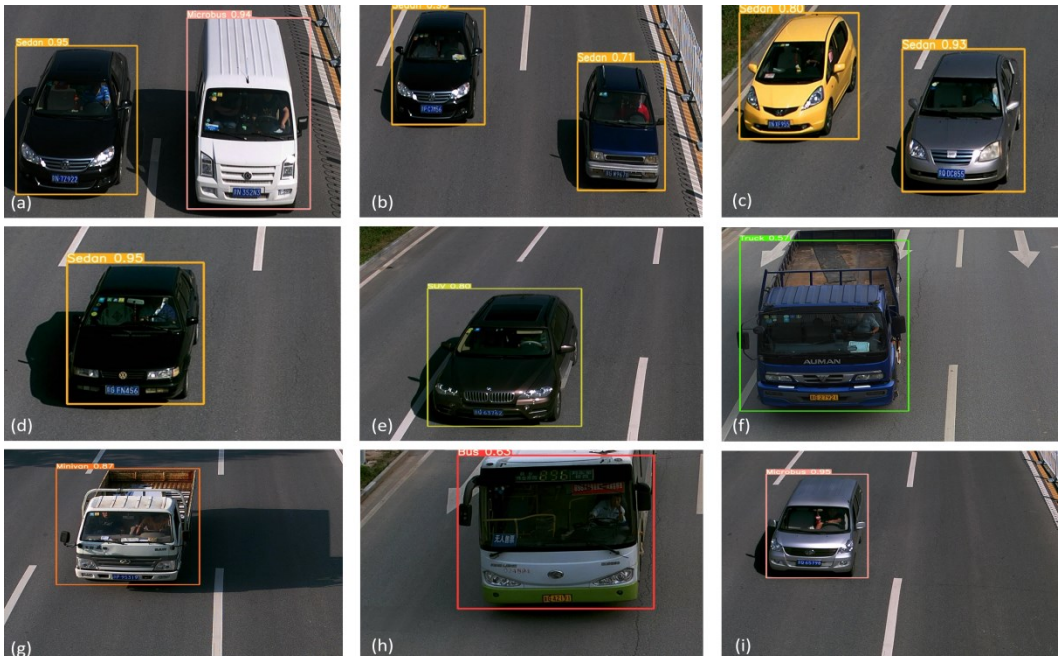

Figure 2. Vehicle detection results of our model.

3.3.2 Generalization Ability of Proposed Network. To check out the generalization ability of the proposed network, we artificially selected some pictures of the CompCars dataset as the new dataset of the paper. Since the dataset is large and has no labels, we selected 1200 images from it as the dataset (Random CompCars Dataset). The result of multiple methods in vehicle detection was presented in Table 5. It is seen from Table 5 that the generalization power of the

proposed algorithm is better than the other algorithms but still performs poorly, with the mAP reaching only 72.9%. The possible reason is that the randomly selected pictures have a low recognition degree. The CompCars dataset only intercepts the front part of the vehicle and only has the information of the "face", resulting in insufficient vehicle information. It is incredibly tough, even literally impossible, for the human eye to distinguish the vehicle type. Figure 3 shows some images of the BIT-Vehicle dataset and the CompCars dataset. And it can be seen from Figure 3 that the two datasets differ significantly.

Table 5. The results of model generalization ability.

Model	mAP (%)
Faster R-CNN+ZF[17]	68.16
Faster R-CNN+VGG16[17]	72.42
Enhanced YOLOv3 tiny network	72.9

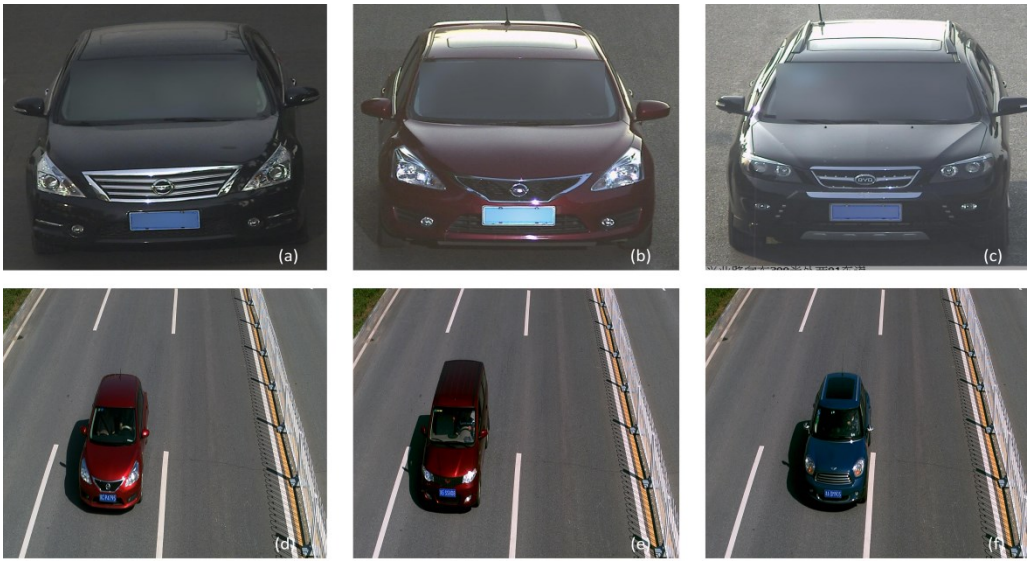

Figure 3. The CompCars dataset is compared with the BIT-Vehicle dataset. (a)-(f): Part of the images in CompCars dataset and BIT-vehicle dataset respectively.

3.3.3 Detection Effect in Harsh Environment Terrible. Bad weather and poor lighting conditions are the two main reasons that affect vehicle detection performance. Figure 4 shows the vehicle detection performance in harsh environments. Figures 4a-4c are the detection results under foggy conditions, Figures 4d-4f are the detection results under the condition of strong light irradiation, and Figures 4g-4i are the detection results under the condition of insufficient light at night test results. As shown in Figure 4 the proposed model can accurately detect vehicles in harsh environments, which is of great significance to driving safety.

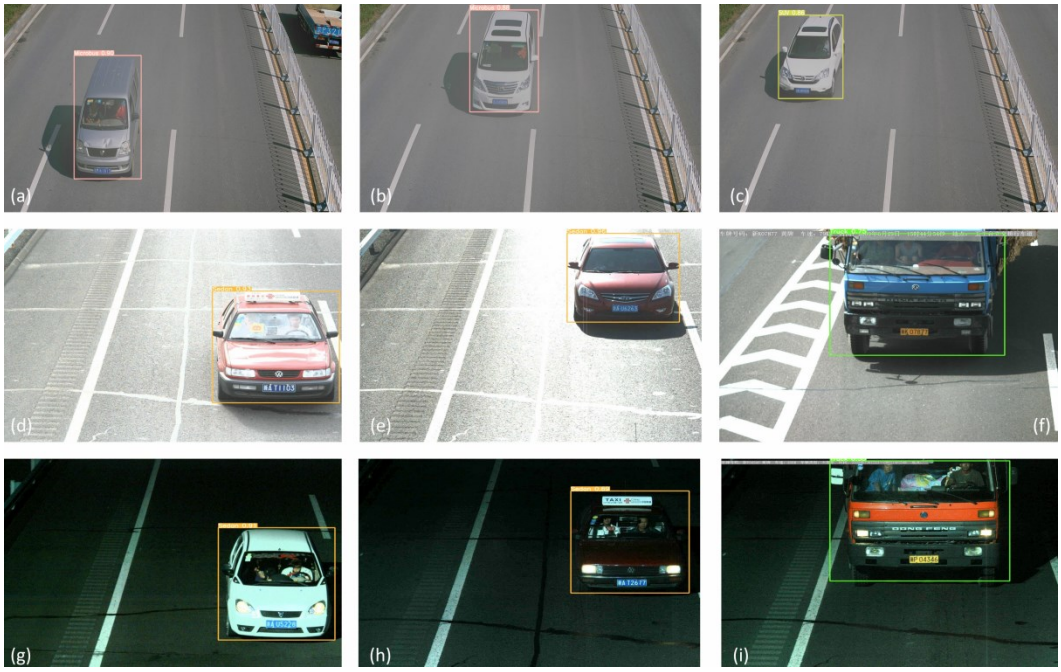

Figure 4. Detection effect under harsh conditions.

4. CONCLUSIONS

We optimize the YOLOv3 tiny network to detect six different types of vehicles in this paper. First, we replaced the Maxpooling layer in the backbone network with the convolutional layer, retaining more target information. Second, we used a dense module in the backbone network, replacing the original residual structure, significantly improving the detection performance of the network. Finally, we retrained the anchor box that is more suitable for vehicle detection to help the network train better. The detection performance of our proposed network is more competitive compared to YOLOv3 and YOLOv3 tiny.

ACKNOWLEDGMENTS

The present study was supported by the High-level Talent Project of Hainan Provincial Natural Science Foundation (grant numbers 620RC557), Hainan Provincial Natural Science Foundation Innovation Research Team Project (grant numbers 620CXTD434), Hainan Provincial Key R&D Plan (grant numbers ZDYF2021GXJS199), National Natural Science Foundation of China and Macau Science and Technology Development Joint Fund (grant numbers 61961160706, 0066/2019/AFJ), and Program of Scientific Research Foundation of Hainan University (KYQD(ZR)1859).

REFERENCES

[1] Wei, Y., Tian, Q., Guo, J., et al., "Multi-vehicle detection algorithm through combining Harr and HOG features," Mathematics and Computers in Simulation, (155), 130-45(2019).
[2] Wang, Z., Zhan, J., Duan, C., Guan, X. and Yang, K., "Vehicle detection in severe weather based on pseudo-visual search and HOG-LBP feature fusion," Proceedings of the Institution of Mechanical Engineers Part D: Journal of Automobile Engineering, 236(7), 1607-18(2021).
[3] Girshick, R., Donahue, J., Darrell, T., et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 580-7(2014).
[4] Girshick, R., "Fast R-CNN," IEEE Inter. Conf. on Computer Vision (ICCV), 1440-8(2015).
[5] Ren, S. Q., He, K. M., Girshick, R. and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 1137-1149(2017).

[6] Redmon, J., Divvala, S., Girshick, R., et al., "You Only Look Once: Unified, real-time object detection," Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 779-88(2016).
[7] Redmon, J. and Farhadi, A., "YOLO9000: Better, faster, stronger," IEEE Conf. on Computer Vision & Pattern Recognition, 6517-25(2017).
[8] Redmon, J. and Farhadi, A., "YOLOv3: An incremental improvement," arXiv:1804.02767, (2018).
[9] Xu, B., Wang, B. and Gu, Y. J., "Vehicle detection in aerial images using modified YOLO," Inter. Conf. on Communication Technology Proc., 1669-72(2019).
[10] Luo, S. J., Xu, C. Y. and Li, H. X., "An application of object detection based on YOLOv3 in traffic image," Inter. Conf. on Image, Video and Signal Processing, 68-72(2019).
[11] Sri Jamiya, S. and Esther Rani, P., "LittleYOLO-SPP: A delicate real-time vehicle detection algorithm," Optik—International Journal for Light and Electron Optics, 225, 165818(2021).
[12] Tajar, A., Ramazani, A. and Mansoorizadeh, M., "A lightweight Tiny-YOLOv3 vehicle detection approach," Journal of Real-Time Image Processing, (18), 2389-401(2021).
[13] Bochkovskiy, A., Wang, C. Y. and Liao, H. Y. M., "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004, (2020).
[14] Hu, X. Z., Wei, Z. and Zhou, W. C., "A video streaming vehicle detection algorithm based on YOLOv4," IEEE Advanced Information Technology, Electronic and Automation Control Conf. (IAEAC), 2081-6(2021).
[15] Glenn, J., https://github.com/ultralytics/YOLOv5, (2020).
[16] Wu, T. H., Wang, T. W. and Liu, Y. Q., "Real-time vehicle and distance detection based on improved YOLOv5 network," 2021 3rd World Symp. on Artificial Intelligence, 24-8(2021).
[17] Sang, J., Guo, P., Xiang, Z. L., et al., "Faster-RCNN model recognition analysis," Journal of Chongqing University, 40(07), 32-6(2017).
[18] Yang, L., Luo, P., Loy, C. C. and Tang, X., "A large-scale car dataset for fine-grained categorization and verification," IEEE Conf. on Computer Vision & Pattern Recognition, 3973-81(2015).