**Vehicle Classification Based on Pulse Coherent Radar**

**Abstract:**

**Key words:**

Intelligent transport system(ITS) is an effective approach to solve the problems such as traffic congestion and difficult parking. The system is based on real-time traffic information detection technology, which include traffic volume, vehicle type and so on. Based on these traffic information obtained accurately, ITS can provide a variety of services for traffic management departments and residents, which include path planning[], autonomous driving[].

The sensors used in the current real-time traffic detection technology mainly include magnetic sensors and cameras. Magnetic sensors have low cost and power consumption, and have a long life cycle, but they are susceptible to magnetic interference from vehicles in adjacent lanes or urban rail transit [3]. Compared with magnetic sensors, the camera can obtain more information, such as the license plate number. But the camera is susceptible to weather and light interference, and the outdoor video detection technology requires the deployment of power lines and communication lines causing the high installation and maintenance costs. At present, the research of radar sensors in the field of intelligent transportation is mostly based on lidar and millimeter-wave radar, and mainly focuses on the field of autonomous driving [4-6]. Lidar and millimeter-wave radar have long detection distances and high accuracy, but they are not suitable for traffic detection in terms of power consumption, size and cost.

The pulse coherent radar, PCR used in this article is a new type of millimeter-wave radar working in the 60GHZ frequency band. It combines the advantages of low power consumption of pulse radar and high accuracy of phase radar[16], with an area of only 29 square millimeters. And it is not interfered by magnetic field and light. Its comparison with lidar and traditional millimeter-wave radar is shown in the table below.

When road vehicle passed above the PCR, the data generated by PCR can reflect the height and profile characteristics of the vehicle chassis, which could be used for vehicle classification. To this end, in this paper, we propose a road vehicle classification approach by deploying PCR in the middle of road lane. In particular, we first design the method to effectively intercept the PCR data when vehicle passed over the PCR. Then we convert each intercepted vehicle data collected in the real road environment into a feature vector of maximum points. Then we use all the feature vectors to fit a Random forest model. The model outputs the type of vehicle according the input data feature vector which is extracted by intercepted vehicle data. Our main contributions are two-fold:

1. . we propose a vehicle classification approach base on the new pulse coherent radar. Design the effective method to intercept the vehicle data and extract the feature of vehicle data. And use the Random forest model to divide the vehicle type into four categories.
2. . Collect large of road vehicle data of PCR in the real road environment. Based on the data, we evaluated three machine learning algorithms SVM[], Random forest[] and CNN[] and concluded that Random forest was the best fit for our problem.

**0 RELATED WORK**

There have been many research on vehicle classification based on different sensors, mainly include magnetic sensor and camera.

**1 PROBLEM DESCRIPTION**

**We use PCR A111 to realize vehicle classification, which provide Envelope working mode supporting high precision ranging and effectively reflecting the** **height and** **profile characteristics of the vehicle chassis.**

The PCR operating in Envelope mode performs one measurement by transmitting a sequence radar pulses and measuring the received pulses energy in different time intervals. The Envelope data generated from one measurement at time t is shown as

ENV(t)=(env1t,envidt,…,envnt) (1)

where ENV(t) is a set of **n** real valued samples ,t refers to the data collected at the t-th time, i refers to sample index and envi(t) **refers to an amplitude reflecting** the received energy from a specific distance which is expressed as

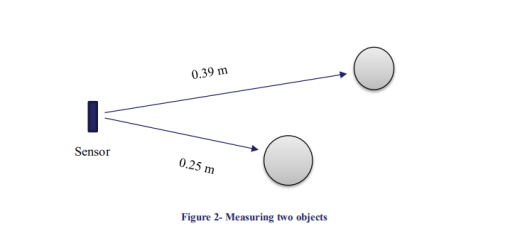
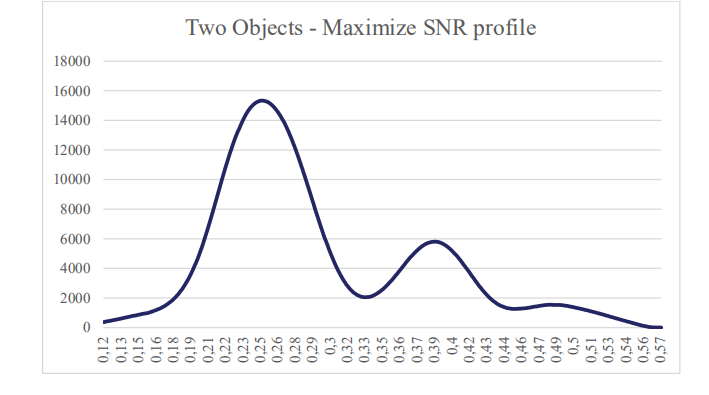
d(i)=res\*i+start (2)

where res is the fixed range resolution which is approximately equal to 0.48mm, start is the closest distance that radar can detect. In addition, there is we have

n=ceil(length/res) (3)

where **length** is the length of the interval that the radar can detect.

For example, when there are two objects near the radar as shown in Fig.1(a), we get the Envelope data generated from one measurement with start of 10cm and length of 40cm shown in Fig.1(b), where we can see there are two peaks at the index 100 and 200. Then we calculate d(100) and d(200) are equal to 8cm and 10cm respectively according to Eq.2. Therefore we estimate that there are two objects at 8cm and 10cm from the radar.

Deploy PCR in the middle of the roadway and assume the vehicle is driving in a lane. Our goal is to get vehicle type when a vehicle passed over the radar. In particular, our problem is divided into two parts.

The first part is vehicle detection to get the start-end time described as follows

(start1,end1,start2,end2,…,)=Detect(ENV(t-N),…,ENV(t)) (4)

where ENV(t-N),…,ENV(t) are the Envelope data collected between time of t-N and t, **t1 and t2 are the start and end time respectively when vehicle passed over the radar**. In addition, the height of vehicle chassis is generated between 15cm and 40cm. Therefore we fix the start and length parameters of radar to 10cm and 40cm. With this configuration and fixed range resolution, the dimension of Envelope data generated from one measurement is 826.

The second part is vehicle classification to get the vehicle type according to the radar signal intercepted by the start-end time

C=Classify(ENV(t1),…,ENV(t2)) (5)

where C is the type of vehicle divided into 5 types{‘moto’, ’sedan’, ‘SUV’, ‘van’, ’bus’}.

PCR

1. **ALGORITHM DESIGN**

A. OVERVIEW

The algorithm overview is shown in Fig.3.

B. VEHICLE DETECTION

The dimension of Envelope data generated from one measurement is 827 and it’s redundant for vehicle detection because the Envelope data changed quite obviously when a vehicle passed over the radar. Therefore we firstly fuse the data by averaging the Envelope data generated from one measurement, shown as

MEnv(t)=mean(env1t,…,env827t) (6)

where MEnv(t) is the averaged data used for subsequent processing.

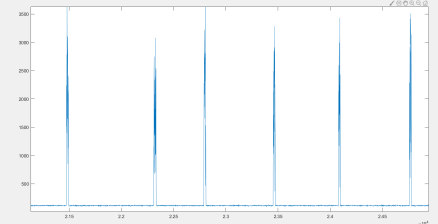
Our algorithm of vehicle detection is divided into 2 steps: 1) According to a dynamic threshold, preliminarily divide the averaged data MEnv(t) into 2 categories: there is vehicle and no vehicle passing over the radar; 2) Process the result of first step to get the start and end time when vehicle passed over the radar by using mathematical morphology.

**1) divide the averaged data MEnv(t) into 2 categories**

Fig.3 shows the averaged data MEnv(t) when 5 vehicles passed over the radar sensor. The data when vehicles pass over the radar is much larger than the data when no vehicles pass by. Therefore, we simply use a threshold to distinguish whether there is a vehicle passing over the radar. In details, we have

|  |  |
| --- | --- |
|  | （7） |

where refers to the dynamical threshold changed by t, S(t)=0 indicates there is no vehicle at the t-th time and S(t)=1 indicates there is a vehicle passing over the radar at the t-th time.



The baseline of data when there is no vehicle will change with the environmental factors such as weather and temperature on the road. Therefore we update the threshold in real time with the baseline which is tracked dynamically by exponential weighted average method. In particularly, we have

thre(t)=(1+namta)b (t) (8)

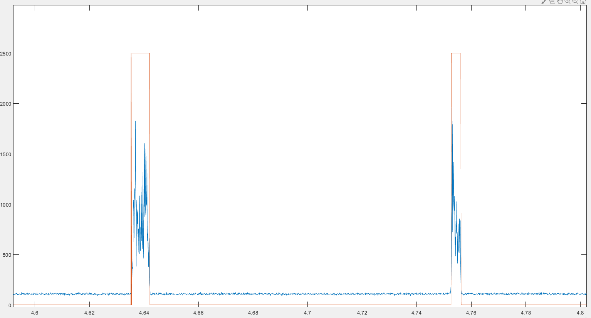
where b(t) is the baseline. Namta is the coefficient to adjust the threshold. The b(t) is updated by

|  |  |
| --- | --- |
|  | （9） |

where w is the weighting factor to update the baseline when S(t)=1.

**2) get the start and end time**

The averaged data MEnv(t) fluctuates greatly when vehicle passed over the radar, and sometimes it is below the threshold. In addition, complex environment on the road makes the Envelope data contain individual noise. Therefore the result from the first step generally has some glitches, shown in Fig.4. Those glitches are not conducive to the calculation of start and end time. To this end, we need to eliminate those glitches before calculate the start and end time..



Our method to eliminate those glitches is based on two operations: corrosion and expansion, which are the basic operations of mathematical morphology[].

Corro(S(t),g)=min{S(t+m)-g(m)} (10)

S(t)&g(n)=max{S(t+m)-g(m)} (11)

where g(m) is the structural parameter. In our scenario, we set

g(m)=0,m=1,…,l (12)

Therefore Eq.10 and 11 is simplified to Eq.13 and 14.

(S(t)^g)(n)=min{S(t+m)} (10)

S(t)&g(n)=max{S(t+m)} (11)

Expansion operation can fill the gully, and corrosion operation can remove the spikes. To eliminate the glitches which appear as gully and spikes, we cascade the two operation shown as Eq.12. We first dilate and erode to fill the gully, then erode and dilate to remove the spikes.

DCCD\_S(t)=DCCD(S(t),5)))) (12)

where DCCD\_S(t) is the filtered result shown in Fig.5.





Then calculate the difference sequence of CDDC\_S(t)

Di\_ CDDC\_S(t)= DCCD\_S(t)- DCCD\_S(t-1),t=1,…

Count the subscripts equal to 1 and -1 in the difference sequence in turn. Finally use these subscripts as the start and end time when vehicle passed over the radar, shown in Fig.6.



We can see the parameter namta, l and w will affect the performance of vehicle detection algorithm. There are some missed judgments based on the dynamical threshold method shown in Eq.x, because the Env(t) changes acutely and sometimes fluctuates below the threshold especially when a bus or truck passed over the radar. Therefore in order to avoid the baseline being incorrectly stretched by averaging these missed data, we set w as 0.2 to ensure the past values of baseline have the much larger weight 0.8 when updating the value of baseline.

The configurations of namta and l are determined by the actual data. To confirm the best values of namta and l, we set different namta and l to calculate the accuracy of vehicle detection on the whole Envelope data we collected with the configuration of PCR in Tab.x, and the result is shown in Fig.x. where we conclude namta 0.3 and l 25 are the best configurations.

In addition, we discovered when namta is bigger and the l should be bigger too to get good performance in the experiment. The threshold becomes bigger when namta is bigger, then there are less incorrect but more missed judgments, which causing more and wider gully in S(t). Therefore the l should be bigger to fill the wider gully.

C. VEHICLE CLASSIFICATION

The Envelope data when vehicle passed over the radar are intercepted by the start-end time for vehicle classification. And the algorithm of vehicle classification is divided into 3 parts: 1) normalization and resize 2) Feature extract; 3) Random forest model[] for vehicle classification.

1. normalization and resize

The original value of Envelope data changes

The speed and length of road vehicles are different, which makes the length of vehicle data collected by radar different. We need to make the length of different vehicle data samples consistent for subsequent classification processing. To keep the information of vehicle data as far as possible, we use bicubic interpolation[] method to adjust each vehicle data sample size to 70\*827, **where 70 is the median of all sample widths.**

2Feature extract

Our vehicle classification algorithm is running in the MCU, which is integrated with the PCR sensor. **To save MCU computing and storage resources, we need to extract features of vehicle data sample.**

The height of vehicle chassis is an effective feature to distinguish different types of cars. However our data is collected during the fast moving of vehicle and the chassis of vehicle is uneven. Therefore it’s difficult to accurately compute the height of chassis.

Fig.7 shows the Envelope data when radar is under SUV. According to the principle of Envelope mode, there is a reflector at the position of the wave crest. Because the chassis of vehicles is uneven. The data has multiple different crests. In other word, the wave crest location of Envelope data is related to the height and profile characteristics of the vehicle chassis.



We extract the maximum points of each Envelope data ENV(t) generated from the t-th measurement as the feature for vehicle classification. In particular, the algorithm of feature extract is divided into 2 steps.

In the first step, we select some candidate points meeting the following loose rule

Envit>Env(i+1)t or Envit>Env(i+2)t

These candidate points should include all maximum points. And the goal of the first step is to avoid missing some maximum points of data ENV(t) by using a loose rule. Fig.8(a) shows the result of first step.

In the second step, we select the maximum points from these candidate points which are distributed over multiple regions shown in Fig.8(a). In particular we select the midpoints of the each region formed by candidate points as the maximum points. Fig.8(b) shows the result of second step. Our algorithm to extract the maximum points can effectively avoid the case of missed selection and multiple selection.



We keep the first three sets of maximum points, and fill them with 0 if there are less than three points. Therefore, each piece of Envelope data is converted into a 6-dimensional vector and the feature size of each vehicle data sample is 70\*6.

1. Random forest model for vehicle classification

In the first step of vehicle classification, we convert vehicle data sample into a feature matrix of size 70\*6. The feature matrix reduces the size of vehicle data sample and can effectively preserves the information of vehicle data sample. Then we categorize the feature matrix into vehicle types based on the machine learning algorithm Random forest.

We evaluated three machine learning algorithms SVM[], Random forest[] and CNN[] and concluded that Random forest was the best fit for our problem. CNN can solve the quite multi-class classification problem to the data of high dimensions and each dimension of data doesn’t need to have specific meaning. The cost of these advantages is that CNN need quite a lot of training data and expensive computing costs. In our problem, we just plan to categorize the feature matrix into 5 types and the feature matrix has relatively small dimensions, which have relatively fixed meaning. Therefore CNN is redundant for our problem. Random forest is a combination of Decision trees and Ensemble learning. The outstanding advantages are simple and fast contributed by the Decision trees, and good generalization performance contributed by Ensemble learning. The performance of SVM and Random forest in our problem is close. But Random forest has a bit higher accuracy and shorter training time than SVM.

In particular, our Random forest model contain 100 decision trees. Each decision tree is fitted in turn by randomly selected **10** dimensions data of feature matrix. The feature matrix are sampled from the whole data set based on the Ensemble learning algorithm Bagging[]. As a result, each decision tree contains 10 decision. With these specific settings and the feature matrix set calculated from the collected vehicle data set, we obtain the Random forest model categorizing the road vehicle into 4 types：car, SUV, bus and truck.

**3 EXPERIMENTS**

In the experiment, the Envelope data has not changed when a vehicle passed by an adjacent lane. Therefore PCR is completely immune to interference from vehicles in adjacent lanes. In fact, even a motorcycles or tricycles passed by the PCR at a very close distance, the Envelope data still has no response. Therefore it’s difficult to distinguish motorcycles or tricycles. For that, our classification task doesn’t include distinguishing motorcycles or tricycles.

Some parameters of PCR are important to the vehicle classification task and the configurations of these parameters are shown in Tab.1. The PCR working in Envelope mode filters each Envelope data by a exponential smoothing filter, which reduce the response of Envelope data when vehicle passed over the radar. Therefore we set the parameter Running ave.fact which is the weight of the filter as 0 to forbidden it. The road vehicle has different length and speed. If the measurement frequency of PCR is too low, it’s unable to detect the vehicle moving too fast. Therefore we set the measurement frequency as 25HZ to ensure there are at least 5 measurements when a vehicle of length 4m and speed 60km/s passed over the radar.

|  |  |
| --- | --- |
| parameter | value |
| Running ave.fact | 0 |
|  | 0.1 |
|  | 0.4 |
|  | 25 |

1. EXPERIMENTAL SETTING
2. EXPERIMENTAL SCENARIO

We collect data on multiple roads in Dongguan, China. As shown in Figure 6, the detection node is deployed in the center of the lane, and the gateway node and the host computer are placed near the detection node and connected through a serial port, and the mobile phone is used to record the vehicle model. The detection node samples at a frequency of 25HZ. The gateway node receives the data of the detection node, and the host computer saves it locally. Finally, 3,000 vehicle data are obtained, including 1,220 cars, 1,035 SUVs, 800 buses, and 330 trucks.

在东莞的多条道路上采集数据。如图6所示，检测节点部署在车道中央，网关节点和上位机放置在检测节点附近并通过串口相连，使用手机记录车型。检测节点以25HZ的频率采样；网关节点接收检测节点的数据，上位机将其保存在本地。最后得到3000条车辆数据，包括1220辆小汽车，1035辆SUV，800辆公交车，330辆货车。



|  |  |
| --- | --- |
| · |  |
|  |  |

1. COMPARISON ALGORITHM

The algorithm proposed in this article is called VC\_PCR\_RF. In order to evaluate Random forest was the better fit for our problem than other machine learning algorithms, two other methods are implemented for comparison experiments:

VC\_PCR\_SVM algorithm: the vehicle classification algorithm of PCR based on SVM. What the difference between VC\_PCR\_SVM and VC\_PCR\_RF is that VC\_PCR\_SVM use the feature matrix set to fit a SVM model rather than a Random forest model in VC\_PCR\_RF.

VC\_PCR\_CNN algorithm: the vehicle classification algorithm of PCR based on CNN. We convert each feature matrix into a square with the size 24\*24. then the square feature matrix set are used to fit a CNN model. The configuration of the model is shown in Tab.x.

|  |  |
| --- | --- |
| CNN\_LATER | parameter |
| Running ave.fact | 0 |
|  | 0.1 |
|  | 0.4 |
|  | 25 |

The 5-fold cross-validation method [20] is used to evaluate the three algorithms. The method is to randomly divide the feature matrix set into 5 equal parts. Choose 4 of them for training, and choose the remaining 1 for testing. Each time a different aliquot is selected for training and testing, and it is executed 5 times in total. Accumulate each test result, and finally get the test result of the algorithm on the entire feature matrix set.

1. PERFORMANCE INDICATORS

The precision and recall are used as the performance indicators to evaluate the proposed vehicle classification algorithm. For a type of vehicles, we define the following concepts to calculate the performance indicators.

True Positive TP: the number of samples belonging to this type and classified as this type.

False Negative FN: the number of samples belonging to other type and classified as this type.

False Positive FP: the number of samples belonging to this type and classified as other type.

True Negative TN: the number of samples belonging to other type and classified as other type.

Then the precision and sensitivity can be calculated by

Accuracy=(TP+TN)/(TP+TN+FP+FN)

precision =TP/(TP+FP)

recall=TP/(TP+FN)

1. EXPERIMENTAL RESULTS

The detail experimental results with the three algorithms are summarized in Tab.7, 8 and 9. It can be seen from the table that the performance of VC\_PCR\_RF

is better than other two algorithms and the algorithm VC\_PCR\_CNN perform worst, which is because we didn’t have quite lager vehicle data to fit the CNN model ,causing the model over fitting.

It can be seen from the table that the algorithm VC\_PCR\_CNN perform worst, which is because we didn’t have quite lager vehicle data to fit the CNN model ,causing the model over fitting. In fact CNN is redundant for our problem.CNN can solve the quite multi-class classification problem to the data of high dimensions and each dimension of data doesn’t need to have specific meaning. The cost of these advantages is that CNN need quite a lot of training data and expensive computing costs. In our problem, we just plan to categorize the feature matrix into 5 types and the feature matrix has relatively small dimensions, which have relatively fixed meaning. Therefore CNN is redundant for our problem.

VC\_PCR\_CNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CAR | SUV | BUS | TRUCK |
| Accuracy | 70% | 71% | 76% | 75% |
| precision | 90% |  |  |  |
| recall |  |  |  |  |

VC\_PCR\_SVM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CAR | SUV | BUS | TRUCK |
| Accuracy | 90% | 88% | 94% | 92% |
| precision |  |  |  |  |
| recall |  |  |  |  |

VC\_PCR\_RF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CAR | SUV | BUS | TRUCK |
| Accuracy | 91% | 92% | 96% | 94% |
| precision |  |  |  |  |
| recall |  |  |  |  |

In the experiment, the performance of SVM and Random forest in our problem is close. But Random forest has a bit higher accuracy than SVM.

The accuracy of car and SUV is relatively low and there are more incorrectly judgments between car and SUV. That’s because the chassis of car and SUV is similar. The bus has the highest accuracy 96%, maybe because the chassis feature of bus is much different from other type vehicles, and the chassis feature of different buses is more similar because the bus model of one city is relatively fixed.

The training time required for three algorithm is shown in Fig.x. we can see CNN need the longest time and the time of RF and SVM is close but RF has the shortest time.

The time required for the algorithm VC\_PCR\_RF to detect and classify different numbers of vehicle samples on MATLAB is shown in Figure 11,and Table 6 lists the detection and classification speeds. The VC\_PCR\_RF

**3.3方法运行效率分析**

三种算法检测不同数量的样本所需的时间如图11所示。MTV用时虽然明显大于ER和MTV-0，但三者的速度都已满足实时性的要求。MTV与MTV-0的主要区别是MTV需要计算扩展特征，而MTV用时明显大于MTV-0，这说明MTV的时间开销主要花在扩展特征的计算上。表6列出了三种算法的检测速度。

****

图11算法检测效率对比图

Fig. 11 Detection efficiency comparison

表6 检测速度

Table6 Detection speed

|  |  |
| --- | --- |
| 检测算法 | 检测单个样本所需的时间 |
| ER | 0.002ms |
| MTV-0 | 0.005ms |
| MTV | 0.040ms |

图12是MTV训练不同数量样本所需的时间，其中各类别的样本比例相同，粒子个数设为100，迭代次数设为50。在上述参数设定下，训练时间与样本数成线性增长，这与本文复杂度分析相符。虽然相比ER算法，MTV需要较耗时的训练过程，但在训练完毕后，MTV的检测速度可以满足实时性要求。



图12 MTV训练不同数量样本的时间

Fig. 12 MTV training time for different numbers of samples

**4 CONCLUSION**

本文提出了一种基于脉冲相干雷达的停车检测算法。提取雷达信号的特征，对雷达信号进行多次预测，结合多次预测的结果实现停车检测。相比已有算法，本文算法提高了干扰场景下停车检测的精度，并且可以检测雷达是否有水或湿树叶的干扰。未来工作包括：1）将本文的方法扩展到更多干扰因素的场景；2）借鉴相邻多个时刻的雷达信号提高检测准确率。

**参考文献**

[1] 易晓珊,李媛红.深圳市智慧路边停车关键技术研究及标准化探索[C]//第十四届中国标准化论坛论文集.中国标准化协会, 2017: 671-677.

YI X S, LI Y H. Key technology research and standardization exploration of smart roadside parking in Shenzhen city[C]. Proceedings of the 14th China Standardization Forum. China Association for Standardization, 2017: 671-677.

[2] Gursel Serpen, Jayanta Debnath. Design and performance evaluation of a parking management system for automated, multi-story and robotic parking structure[J]. International Journal of Intelligent Computing and Cybernetics, 2019, 12(4): 444-465.

[3] 关向科.基于无线磁阻传感器网络的车辆检测技术研究[D].南京:南京邮电大学, 2014.

GUAN X K. Research on vehicle detection technology based on wireless magnetoresistive sensor network[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2014.

[4] Ding J, Cheung S Y. Signal processing of sensor node data for vehicle detection［C］/ /The 7th International IEEE Conference on Intelligent Transportation Systems， 2004: 70-75.

[5] 张足生,邓见光,赵铁柱,等.一种波形相似度的车辆检测融合算法[J].传感技术学报, 2018, 31(3): 400-407.

ZHANG Z S, DENG J G, ZHAO T Z, et al. A fusion algorithm for vehicle detection based on waveform similarity[J]. Chinese Journal of Sensors and Actuators, 2018, 31(3): 400-407.

[6] 李明熹,林正奎,曲毅.计算机视觉下的车辆目标检测算法综述[J].计算机工程与应用, 2019, 55(24): 20-28.

LI M X, LIN Z K, QU Y. Survey of vehicle object detection algorithm in computer vision[J]. Computer Engineering and Applications, 2019, 55(24): 20-28.

[7] 毛其超,贾瑞生,左羚群,等.基于深度学习的交通监控视频车辆检测算法[J].计算机应用与软件, 2020, 37(9): 111-117.

MAO Q C, JIA R S, ZUO L Q, et al. A traffic surveillance video vehicle detection method based on deep learning[J]. Computer Applications and Software, 2020, 37(9): 111-117.

[8] Ren S, He K, Girshick R, et al. FasterR-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6): 1137-1149.

[9]顾恭,徐旭东.改进YOLOv3的车辆实时检测与信息识别技术[J/OL].计算机工程与应用 (2020-08-31) [2020-10-29]. http://kns.cnki.net/kcms/detail/11.2127.TP.20200831.1435.006.html.

GU G, XU X D. Real-time vehicle detection and information recognition technology based on YOLOv3 improved algorithm. [J/OL]. Computer Engineering and Applications: 1-17[2020-10-20]. http://kns.cnki.net/kcms/detail/11.2127.TP.20200831.1435.006.html.

[10] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 379-387.

[11] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017: 6517-6525.

[12] Redmon J, Farhadi A. YOLOv3: an incremental improvement[J]. arXiv: 1804. 02767, 2018.

[13] Rui Fu, Mingfang Zhang, Chang Wang. Behavior analysis of distant vehicles using LIDAR point cloud[J]. Cluster Computing, 2019, 22(2): 8613-8622.

[14] Eum, Bae, Jeon, et al. Vehicle detection from airborne LiDAR point clouds based on a decision tree algorithm with horizontal and vertical features[J]. Remote Sensing Letters, 2017, 8(5): 409-418.

[15] C. H. Jang, C. S. Kim, K. C. Jo, et al. Design factor optimization of 3D flash lidar sensor based on geometrical model for automated vehicle and advanced driver assistance system applications[J]. International Journal of Automotive Technology, 2017, 18(1): 147-156.

[16] Acconeer AB. Radar sensor introduction[EB/OL] (2020-09-29) [2020-10-29]. https://acconeer-python-exploration.readthedocs.io/en/latest/sensor\_introduction.html.

[17] 杨维,李歧强.粒子群优化算法综述[J].中国工程科学, 2004, 6(5): 87-94.

YANG W, LI Q Q. Summary of particle swarm optimization algorithms[J]. Strategic Study of CAE, 2004, 6(5): 87-94.

[18] 丁小欧,于晟健,王沐贤,等.基于相关性分析的工业时序数据异常检测[J].软件学报, 2020, 31(3): 726-747.

DING X O, YU S J, WANG M X, et al. Anomaly detection on industrial time series based on correlation analysis[J]. Journal of Software, 2020, 31(3): 726-747.

[19] 雷蕾,王晓丹,罗玺,等.ECOC多类分类研究综述[J].电子学报, 2014, 42(9): 1794-1800.

LEI L, WANG X D, LUO X, et al. An overview of multi-classification based on error-correcting output codes[J]. Acta Electronica Sinica, 2014, 42(9): 1794-1800.

[20] 杨柳,王钰.泛化误差的各种交叉验证估计方法综述[J].计算机应用研究, 2015, 32(5): 1287-1290.

YANG L, WANG Y. A review of various cross-validation estimation methods for generalization errors[J]. Application Research of Computers, 2015, 32(5): 1287-1290.

