**Vehicle Classification Based on Pulse Coherent Radar**

**Abstract:** The technology of traffic information collection is the basis of Intelligent transport system. However it’s a challenge to effectively perform the road vehicle classification, due to the dynamical traffic environment and various types of vehicle on the road. In this paper, we proposed a real time approach of road vehicle classification based on the new Pulse Coherent Radar, which is deployed in the middle of road lane. We first intercept the vehicle data when a vehicle passes over the radar. Then extract the feature from the intercepted vehicle data to fit a Random Forest model. According to the input data of feature, the model output the type of vehicle, which include car, SUV, bus and middle truck. In the experiment, we collected the sufficient vehicle data in the actual road environment, and the average accuracy of our approach is 93%.

**Key words:** Internet of Things, intelligent transportation system, vehicle classification, pulse coherent radar, random forest.

**I. INTRODUCTION**

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. Based on these traffic information obtained accurately, ITS can provide a variety of services for traffic management departments and residents, which include path planning [1]-[3], autonomous driving [4]-[7].

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 [8]. 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 [9]-[11]. 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 [12], with an area of only 29 square millimeters. And it is not interfered by magnetic field and light.

When road vehicle passes 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 passes 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 collected data, we evaluated the proposed approach, which shows the average accuracy is 93%.

**II. RELATED WORK**

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

In [13], a group of magnetic sensors are placed along the roadside for vehicle detection and classification, where vehicles are classified into four groups by estimating their magnetic length. In [14], a single three-axis magnetic sensor is deployed along the roadside. The magnetic field data of each vehicle is converted into 2-dimensional images and the vehicle is categorized into 7 types by a 2-dimensional convolution neural network (CNN). In [15], the authors extract the features of relative vehicle length, total waveform energy, and "peak-valley graph", then use hierarchical decision tree algorithm to perform vehicle classification, which is suitable for embedded systems because of the small amount of calculation.

With the development of artificial intelligence, the research of vehicle classification based on camera increasingly focus on deep learning algorithms include Faster R-CNN [16]-[17], SSD [18] and YOLO [19]-[21]. In [22], the authors present a novel method for vehicle detection based on the MobileNet which is integrated into Faster R-CNN structure. The method improves the detection accuracy and saves computation resources compared with Faster R-CNN. In [23], the authors propose a real-time system to enhance the accuracy level on detection and classification of vehicles for a multi-view surveillance video using an optimized YOLOv2 deep learning algorithm.

Although there have been many research of vehicle classification based on magnetic sensor or camera. It’s always difficult to solve the interference problems of magnetic sensor and camera. And the previous radars lidar and millimeter-wave radar are not suitable for traffic information collection because of the power consumption, size and cost. Therefore, there is the important value of road vehicle classification research based on the new radar PCR, which is not interfered by magnetic field, sunlight and weather and has the advantages of low power consumption, small size and low cost.

**III. PROBLEM DESCRIPTION**

Deploy PCR in the middle of the roadway and assume the vehicle is driving in a lane. When road vehicle passes over the PCR, the data generated by PCR can reflect the height and profile characteristics of the vehicle chassis for vehicle classification. The model A111 of radar PCR is used in our scenario. A111 provides Envelope mode which supports high precision ranging and can reflect the height and profile characteristics of the vehicle chassis.

The A111 working 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 the *t*-th time is shown as

|  |  |
| --- | --- |
| , | (1) |

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

|  |  |
| --- | --- |
| , | (2) |

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

|  |  |
| --- | --- |
| , | (3) |

where is the length of the distance interval that the radar can detect. Eq. 3 indicates the number of samples *n* is determined by the parameter .

When there are two objects near the radar as shown in Figure 1(a), we get the Envelope data generated from one measurement with of 10 cm and of 40 cm shown in Figure 1(b), where we can see there are two peaks at the sample counts of 200 and 416. Then we calculate and are approximately equal to 20 cm and 30 cm respectively according to Formula 2. Therefore we estimate that there are two objects at 20 cm and 30 cm from the radar.

|  |  |
| --- | --- |
|  |  |
| (a) Measurement scene | (b) Envelope data |
| Fig. 1 Envelope data generated by one measurement | |

Our goal is to obtain vehicle type when a vehicle passes over the radar. In particular, our problem is divided into two parts.

The first part is vehicle detection to get the times of measurement when the vehicle is driving towards and away from the radar which is called “start-end times” in this paper. The first part is described as follows

|  |  |
| --- | --- |
| , | (4) |

where the input data are the Envelope data collected between -th time and-th time measurements.The output data are the measurement times of the *i*-th vehicle driving towards and away from the radar between the -th and -th measurements, respectively.

The second part is vehicle classification to obtain the vehicle type according to the Envelope data intercepted by each pair of start-end times

|  |  |
| --- | --- |
| , | (5) |

where is the type of the *i*-th vehicle divided into four types .

**IV. ALGORITHM DESIGN**

A. OVERVIEW

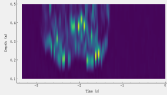
The overview of the approach we proposed is shown in Figure 2. The original data are collected from the radar PCR, which is deployed in the middle of road lane. Then the module of vehicle detection effectively intercepts the data when vehicle passes over the radar. Then the intercepted vehicle data are adjusted to a fixed size. Then we extract the feature from the processed vehicle data. With the input data of feature vector, the trained Random Forest model output the result of vehicle type.

Vehicle

Detection



Original data



Feature Extract

Feature vector

Random Forest Model

Vehicle type

Resize

Intercepted vehicle data

Fig. 2 Approach overview

B. VEHICLE DETECTION

Figure 3 shows the Envelope data of SUV and bus passing over the radar. The Envelope data generated from one measurement has too much samples with the number *n*, which are redundant for vehicle detection because the Envelope data changes quite obviously when the vehicle passes over the radar as shown in Figure 3.



Fig. 3 Envelope data of vehicle passing over the radar

Therefore we firstly fuse the data by averaging the Envelope data generated from one measurement, expressed as

|  |  |
| --- | --- |
| , | (6) |

where is the averaged Envelope data generated from the *t*-th time measurement.

Our algorithm of vehicle detection is divided into 2 steps: 1) Preliminarily divide the averaged data into 2 categories: there is vehicle or no vehicle passing over the radar, according to a dynamic threshold. 2) Further calculate the start-end times through open-close operation of Mathematical Morphology [24].

**Step 1: divide the averaged data into 2 categories**

Figure 4 shows the averaged data calculated from the Envelope data in Figure 3. The when the vehicle passes the radar is much larger than the when no vehicle passes 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*, indicates there is no vehicle at the *t*-th time measurement and indicates there is a vehicle passing over the radar at the *t*-th time measurement.



Fig. 4 Averaged data of vehicle passing over the radar

The baseline of data when there is no vehicle will changes 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

|  |  |
| --- | --- |
| , | (8) |

where is the baseline. is the coefficient to adjust the threshold. The is updated by

|  |  |
| --- | --- |
|  | （9） |

where is the weighting factor to update the baseline when .

**Step 2: calculation of the start-end times**

The averaged data fluctuate 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, which appear as gully and spikes shown in Figure 5. Those glitches are not conducive to the calculation of start-end times. To this end, we need to eliminate those glitches before calculating start-end times.



Fig. 5 Result of step 1

Our method to eliminate those glitches is based on two operations: corrosion and expansion, which are the basic operations of Mathematical Morphology.

|  |  |
| --- | --- |
| , | (10) |

|  |  |
| --- | --- |
| , | (11) |

where is the structural parameter, is the length of . and are the results obtained by respectively corroding and expanding with the structural . In our scenario, we set

|  |  |
| --- | --- |
| , | (12) |

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

|  |  |
| --- | --- |
| , | (13) |

|  |  |
| --- | --- |
| , | (14) |

Then the open and close operation are realized by combining the two operations of corrosion and expansion.

|  |  |
| --- | --- |
| , | (15) |

|  |  |
| --- | --- |
| , | (16) |

where and refer to open and close operation respectively.

The close operation can fill the gully, and the open operation can remove the spikes. To calculate the start-end times, we first perform the close operation to fill the gully, then perform the open operation to remove the spikes, which is called close-open operation expressed as

|  |  |
| --- | --- |
| , | (17) |

where is the filtered result of performing close-open operation on . The filtered result is shown in Figure 6.



Fig. 6 Filtered result

After obtaining the filtered result, we calculate the difference sequence of CO\_S

|  |  |
| --- | --- |
| , | (18) |

Finally, record the subscripts equal to 1 corresponded to the start time and -1 corresponded to the end time in the difference sequence in turn, and use these subscripts as the start-end times of different vehicles.

C. VEHICLE CLASSIFICATION

**After performing the vehicle detection, intercept the Envelope** data by the start-end times and each intercepted data is a vehicle sample. The obtained vehicle samples of SUV and bus are shown in Figure 7, where the heights of the vehicle samples are corresponded to the total number of samples n, and the widths of the vehicle samples are corresponded to the total number of measurement.

After obtaining the vehicle sample, the algorithm of vehicle classification is divided into 3 steps: 1) Resize; 2) Feature extract; 3) Random forest model [25] for vehicle classification.



Fig. 7 Vehicle samples of SUV and bus

**Step 1: Resize**

The speed and length of road vehicles are different, which makes the widths of the vehicle samples different. We need to make the size of different vehicle samples consistent for subsequent classification processing. Because the height of each vehicle sample is fixed as *n*, therefore we just need to adjust the width of sample.

In order to save storage and computing resources, we fix the width as the median of all sample widths, instead of the maximum of all sample widths. In particular, when the width of sample is smaller than the median, we use Cubic Spline Interpolation [26] on each row of the sample to expand the vehicle sample size to , where is the median of all sample widths. Otherwise we downsample each row of the sample to reduce the size to .

**Step 2: Feature extract**

The algorithm of vehicle classification needs to run on embedded devices which is integrated with the PCR. Extract the effective feature from vehicle sample for subsequent processing, which helps to save the computing and storage resources of the embedded devices.

Figure 8 shows the three pieces of Envelope data of the vehicle sample of SUV, where there are multiple different crests in each Envelope data. According to the principle of Envelope mode, there is a reflector at the position of the wave crest. Because our data is collected during the fast moving of vehicle and the chassis of vehicle is uneven. Therefore the Envelope data of one measurement in the vehicle sample generally has multiple crests, and the wave crest location of Envelope data is related to the height and profile characteristics of the vehicle chassis.



Fig. 8 Envelope data generated from different measurement times

Although the height of vehicle chassis is an effective feature to distinguish different types of cars, it’s difficult to accurately compute the height of chassis because of the multiple crests in the Envelope data. In our scheme, we extract the wave crest of each one Envelope data in the vehicle sample as the features of vehicle sample. The wave crest is expressed as the maximum points , where the is the index of samples and the is the amplitude of the x-th sample.

Figure 9 shows the wave crest of the Envelope data of vehicle entering, where the wave crest is not strictly smooth. In fact, the Envelope data collected in actual environment is discrete and the wave crest of the data can’t strictly conform the mathematical definition of maximum points in many cases.



Fig. 9 Wave crest of the Envelope data of vehicle entering

For that, the calculation of the maximum points of the Envelope data (generated from the *t*-th measurement) is divided into 2 steps.

In the first step, we select some candidate points from all the sample points in . These candidate points meet a loose condition, which is that for the 9 points before the candidate point, the y of the candidate point is at least larger than one of them, and for the 9 points behind the candidate point, the y of the candidate point is also at least larger than one of them.

The significance of the first step is to avoid missing some maximum points of by using a loose condition. And these candidate points should include all maximum points. Figure 10 shows the result of first step.



Fig. 10 Candidate points selected from

In the second step, we select the maximum points from these candidate points which are distributed over multiple regions shown in Figure 10. In particular we select the midpoints of each region formed by candidate points as the maximum points. Figure 11 shows the result of the second step. Our algorithm to extract the maximum points can effectively avoid the case of missed selection and multiple selection.



Fig. 11 Maximum points selected from

To fix the size of the feature extracted from one Envelope data, we only 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.

Perform the calculation of maximum points to each one Envelope data in the vehicle sample. Eventually we obtain the feature vector of size extracted from the vehicle sample.

**Step 3: Random forest model for vehicle classification**

In the first two steps of vehicle classification, we have converted the vehicle sample into the feature vector of size . In the third step, we categorize the feature vector into vehicle types based on the machine learning algorithm Random forest [26] which has the advantages of simple, fast and good generalization performance.

In our scenario, the Random forest model is trained by the feature vector set obtained from the whole vehicle samples collected in the actual environment. The model contains 100 decision trees, and each decision tree is trained in turn by a subset of the feature vector set. After training, we obtain the Random forest model categorizing the road vehicle into 4 types：car, SUV, bus and truck.

**V. EXPERIMENTS**

1. EXPERIMENTAL SETTING

Some parameters of PCR are important to the vehicle classification task and the configurations of these parameters are shown in Table 1.

The height of road vehicles chassis is generally between 15 cm and 40 cm. Therefore we fix the parameters and to 10 cm and 40 cm. With this configuration and fixed range resolution, the dimension of Envelope data generated from one measurement is 826.

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 weight of the filter which is the parameter average fact 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 25 HZ to ensure there are at least 5 measurements when a vehicle of length 4 m and speed 70 km/h passes over the radar.

Tab. 1 Experimental parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Value |
|  | the closest distance that PCR can detect | 0.1 |
|  | the length of the distance interval that PCR can detect | 0.4 |
|  | measurement frequency of PCR | 25 |
| average fact | weight of the exponential smoothing filter | 0 |

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.

With the configurations of Table 1, 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 middle-trucks.

|  |
| --- |
| · |
| Fig. 12 Experimental scenario |

B. PARAMETER CONFIGURATION OF VEHICLE DETECTION ALGORITHM

The parameters , and in Formulas 8, 9 and 10 will affect the performance of the vehicle detection algorithm.

As shown in Figure 4, the averaged data changes acutely and sometimes fluctuates below the threshold especially when a bus passed over the radar. Therefore there are some missed judgments based on the method shown in Formula 7. In order to avoid the baseline being incorrectly stretched by averaging these data of missed judgments, we set as 0.2 to ensure the past values of baseline have the much larger weight 0.8 when updating the value of baseline.

There is a correlation between and , which is that when  is bigger and the should be bigger too to get good performance in general. Because the threshold becomes bigger when is bigger, then there are less incorrect but more missed judgments, which causing more and wider gully in , then the should be bigger to fill the wider gully. The configurations of and are determined by the actual data. To confirm the best values of and , we set different and to calculate the accuracy of vehicle detection on the whole collected data, and the result is shown in Figure 13, where we conclude 0.2 and 17 are the best configurations.

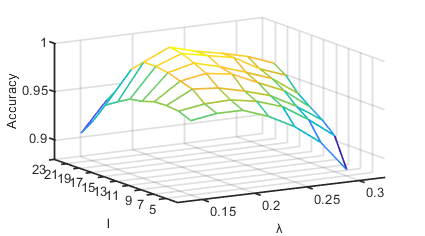


Fig. 13 Accuracy of vehicle detection with different l and

C. ACCURACY OF VEHICLE CLASSIFICATION

1. COMPARISON ALGORITHM

The approach proposed in this article is called VCRF. Two other methods are implemented for comparison experiments:

VCSVM: the vehicle classification algorithm based on SVM. What the difference between VCSVM and VCRF is that VCSVM use the feature vector set to fit a SVM model rather than a Random forest model in VCRF.

VCCNN: the vehicle classification algorithm based on CNN. We convert each feature vector into a square with the size 24\*24. Then the square feature vector set are used to fit a CNN model. The parameters of the CNN model are shown in Tab.2.

Tab. 2 Parameters of CNN model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer | Kernel size | Kernel number | Stride | Input size |
| Conv2D | 33 | 8 | 1 |  |
| Conv2D | 33 | 16 | 1 |  |
| Maxpool | 22 | / | 2 |  |
| Conv2D | 33 | 32 | 1 |  |
| Full | / | 4 | / |  |

The 5-fold cross-validation method [20] is used to evaluate the three algorithms. The method is to randomly divide the feature vector 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 vector set.

1. PERFORMANCE INDICATORS

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 performance indicators accuracy can be calculated by

3) ACCURACY RESULTS

The detail experimental results with the three algorithms are summarized in Table 3, where the performance of VCRF is better than other two algorithms. According to the classification results by the algorithm VCRF, the accuracy of car and SUV is relatively low, that’s because the chassis of car and SUV is similar, causing more incorrectly judgments between car and SUV. The bus has the highest accuracy 96%, 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.

Tab. 3 Accuracy of vehicle classification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Car | SUV | Bus | Middle-Truck |
| VCRF | 90% | 92% | 96% | 94% |
| VCSVM | 90% | 88% | 94% | 93% |
| VCCNN | 69% | 71% | 76% | 75% |

The performance of VCSVM and VCRF in our problem is close, but VCSVM has a bit lower accuracy than VCRF.

The algorithm VCCNN 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 the 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 vector into 5 types and the feature matrix has relatively small dimensions, which have relatively fixed meaning. Therefore CNN is redundant for our problem.

D. SPEED OF VEHICLE CLASSIFICATION

Figure 14 shows the time required for the algorithms VCRF and VCSVM to classify different numbers of vehicle samples, where VCRF requires more time. Table 4 lists the speeds of the two algorithms to classify a single vehicle sample. The speed of VCRF can meet real-time requirements, although it is slightly lower than VCSVM.



Fig. 14 Classification efficiency comparison

Tab. 4 Classification speed

|  |  |
| --- | --- |
| Algorithm | Time required to classify one vehicle sample |
| VCRF | 51.6ms |
| VCSVM | 46.5ms |

**VI. CONCLUSION**

The road vehicle classification is the basis of ITS. In this paper, we have proposed a road vehicle classification approach based on the new radar sensor, PCR. In the approach, we first intercept the vehicle data effectively by a dynamical threshold and open-close operation, which can effectively deal with the individual noise in actual environment. Then design a robust method (the main principle is to choose twice) to extract the feature vector of maximum points from the intercepted vehicle data, which is used as the input of a Random Forest model. The model which has been trained by the feature vectors calculated by all the collected data in the actual road environment, categorizes the vehicle into 4 types. The experimental result has shown the averaging accuracy of our approach is 93%.

In the future works, we will realize the traffic information collection system based on the radar PCR, where the radar node will embed the vehicle classification approach from this paper and parking detection approach from [27].

**REFERENCES**

[1] W. Gong, B. Zhang, and C. Li, “Location-based online task assignment and path planning for mobile crowdsensing,” IEEE Trans. Veh. Technol., vol. 68, no. 2, pp. 1772–1783, Feb. 2019.

[2] B. Irani, J. Wang, and W. Chen, “A localizability constraint-based path planning method for autonomous vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 7, pp. 2593–2604, Jul. 2019.

[3] C. Guo, D. Li, G. Zhang, and M. Zhai, “Real-time path planning in urban area via VANET-assisted traffic information sharing,” IEEE Trans. Veh. Technol., vol. 67, no. 7, pp. 5635–5649, Jul. 2018.

[4] H. Peng, Q. Ye, and X. S. Shen, “SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach”, IEEE Wireless Commun., vol. 26, no. 4, pp. 156 162, Aug. 2019.

[5] Q. Luo, Y. Cao, J. Liu, and A. Benslimane, “Localization and navigation in autonomous driving: Threats and countermeasures,” IEEE Wireless Commun., vol. 26, no. 4, pp. 38 45, Aug. 2019.

[6] Z. Su, Y. Hui, and T. H. Luan, “Distributed task allocation to enable collaborative autonomous driving with network softwarization,” IEEE J. Sel. Areas Commun., vol. 36, no. 10, pp. 2175 2189, Oct. 2018.

[7] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. Shen, “Performance analysis of IEEE 802.11p DCF for multiplatooning communications with autonomous vehicles,” IEEE Trans. Veh. Technol., vol. 66, no. 3, pp. 2485 2498, Mar. 2017.

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

[9] Rui Fu, Mingfang Zhang, Chang Wang, “Behavior analysis of distant vehicles using LIDAR point cloud,” Cluster Computing, vol. 22, no. 2, pp. 8613-8622, 2019.

[10] Eum, Bae, Jeon, et al, “Vehicle detection from airborne LiDAR point clouds based on a decision tree algorithm with horizontal and vertical features,” Remote Sensing Letters, vol. 8, no. 5, pp. 409-418,. 2017.

[11] 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,” International Journal of Automotive Technology, vol. 18, no. 1, pp. 147-156,. 2017.

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

[13] W. Balid, H. Tafish, and H. H. Refai, “Intelligent vehicle counting and classification sensor for real-time traffic surveillance,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 6, pp. 1784 1794, Jun. 2018.

[14] Li, Wengang, Liu, Zhen, Hui, Yilong, Yang, Liuyan, Chen, Rui; Xiao, Xiao (2020), “Vehicle Classification and Speed Estimation Based on a Single Magnetic Sensor,” IEEE Access, pp. 147-156,. 2020.

[15] Kaewkamnerd S, Chinrungrueng J, “Vehicle classification based on magnetic sensor signal,” IEEE International Conference on Information and Automation (ICIA), 2010, pp. 935-939.

[16] GIRSHICK R, “Fast R-CNN,” Proceedings of 2015 IEEE International Conference on Computer Vision, 2015, pp. 1440-1448.

[17] REN S, HE K, GIRSHICK R, et al, “Faster R-CNN: towards real-time object detection with region proposal networks,” International Conference on Neural Information Processing Systems, 2015, pp. 91-99.

[18] LIUW, ANGUELOV D, ERHAN D,et al, “SSD: single shot multibox detector,” European Conference on Computer Vision, 2016, pp. 21-37.

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

[20] Redmon J, Farhadi A, “YOLO9000: better, faster, stronger,” Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017: pp. 6517-6525.

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

[22] Wentao LYU et al, “Vehicle Detection Based on an Imporved Faster R-CNN Method:Regular Section,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, pp. 587-590,. 2021.

[23] N. Kavitha and D.N. Chandrappa, “Optimized YOLOv2 based vehicle classification and tracking for intelligent transportation system,” Results in Control and Optimization, 2021, 2.

[24] Nahrawi Nadzirah et al, “A Novel Nucleus Detection on Pap Smear Image Using Mathematical Morphology Approach,” Journal of Biomimetics, Biomaterials and Biomedical Engineering 6290(2021): pp.53-61.

[25] Tao Chen et al, “Object-Oriented Landslide Mapping Using ZY-3 Satellite Imagery, Random Forest and Mathematical Morphology, for the Three-Gorges Reservoir, China,” Remote Sensing 9.4(2017).

[26] Blatov I A and Dobrobog N V and Kitaeva E V, “The cubic interpolation spline for functions with boundary layer on a Bakhvalov mesh,” Journal of Physics: Conference Series 1715.1(2021): 012001-.

[27] ZHANG X C, ZHANG Z S, LI Y G, GUO X H, CHEN W, “Vehicle detection algorithm based on pulse coherent radar,” Computer Engineering and Applications. doi:10.3778/j.issn.1002-8331.2011-0032.