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. We deploy the radar in the middle of road lane and intercept the vehicle data when a vehicle passes over the radar. Then extract the feature of vehicle chassis outline and height from the intercepted vehicle data to fit a Random Forest model. According to the input data of features, 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 94.03%.

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 the traffic information obtained accurately, ITS can provide a variety of services for traffic management departments and residents, e.g., 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 60 GHZ 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 mm^2, 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 outline and height of the vehicle chassis, which could be used for vehicle classification. To this end, 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 vehicle chassis outline and height. Finally, we use all the feature vectors to fit a Random forest model, and the model divides the road vehicle into four types: car, SUV, bus and middle-truck. The contributions of this paper 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 features of vehicle chassis outline and height. And use the Random forest model to divide the vehicle type into four categories.

2) Collect sufficient vehicle data in the actual

environment. Based on the collected data, we evaluated the proposed approach, which shows the average accuracy is 94.02%.

The rest of the paper is organized as follows. Section II provides related work. Section III introduces the radar PCR and describes the classification task based on PCR. Section IV details the proposed approach of vehicle detection and classification. Section V evaluates the approach based on the data collected in actual environment, followed by conclusion and future work in Section VI.

II. RELATED WORK

There have been many studies 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, and the magnetic field data of each vehicle is converted into 2-dimensional image 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 focuses on deep learning algorithms, e.g., 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 studies 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, e.g., 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 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 [12].

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 outline and height of the vehicle chassis for vehicle classification.

The PCR model A111, which is used in our scenario, provides Envelope mode that supports high precision ranging. And 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 the *t*-th measurement is shown as

$$ENV(t) = (env_1(t),...,env_i(t),...,env_n(t)), \quad (1)$$

where $ENV(t)$ is a set of *n* real valued samples, *t* refers to that the data are collected at the *t*-th measurement, $env_i(t)$ refers to an amplitude representing the received energy from a specific distance $d(i)$ which is calculated by

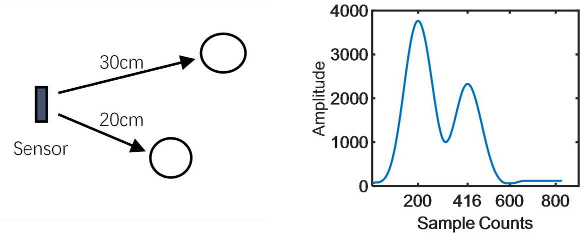
$$d(i) = \Phi_{res} i + d_{start}, \quad (2)$$

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

$$n = \frac{d_{length}}{\Phi_{res}}, \quad (3)$$

where d_{length} is the range length of the radar detection. Equation 3 indicates the number of samples *n* is determined by the parameter d_{length}.

When there are two objects near the radar shown in Figure 1(a), we get the Envelope data generated from one measurement shown in Figure 1(b), where the d_{start} and d_{length} are equal to 10 cm and 40 cm. We can see there are two peaks at the sample counts of 200 and 416, then we calculate $d(200)$ and $d(416)$ are approximately equal to 20 cm and 30 cm respectively according to Equation 2, which indicating the received energy at 20 and 30 cm from the radar is larger. Therefore, we estimate that there are two objects at 20 cm and 30 cm from the radar, respectively.



(a) Measurement scene (b) Envelope data

Fig. 1 Envelope data generated by one measurement

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

The first part is vehicle detection to get the measurement counts when the vehicle is entering and leaving the radar. The first part is described as follows

$$\begin{aligned}&(enter_1,leave_1,\ldots,enter_i,leave_i,\ldots)\\&=Detect(ENV(t_0),ENV(t_1),\ldots,ENV(t_s)),\end{aligned}\qquad(4)$$

where the input $ENV(t_0),ENV(t_1),\ldots,ENV(t_s)$ are the Envelope data collected between the t_0-th and t_s-th measurement, and the output $enter_i$, $leave_i$ are the measurement counts of the i-th vehicle entering and leaving the radar, respectively.

The second part is vehicle classification to obtain the vehicle types between the t_0-th and t_s-th measurement, described as follows

$$\begin{gathered}C_i=Classify(ENV(enter_i),\ldots,ENV(leave_i)),\\i=1,2,\ldots\qquad(5)\\C_i\in\{\text{‘car’, ‘SUV’, ‘bus’, ’middle truck’}\},\end{gathered}$$

where the input $ENV(start_i),\ldots,ENV(end_i)$ are the Envelope data when the i-th vehicle passes the radar, and the output C_i is the i-th vehicle type, which may be car, SUV, bus, and middle-truck.

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 resized to a fixed size. Then we extract the feature vector of vehicle chassis outline and height from the resized vehicle data. With the input data of feature vector, the trained Random Forest model output the result of vehicle type.

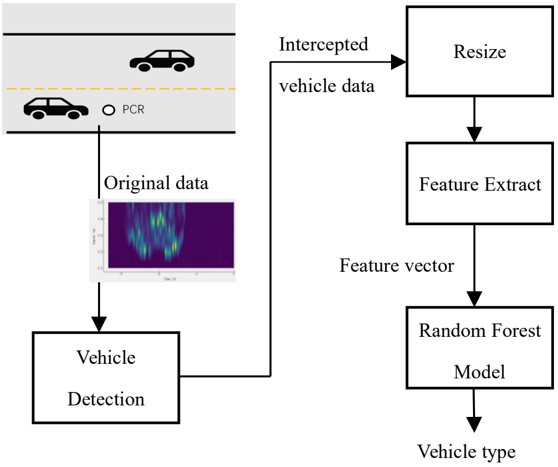


Fig. 2 Approach overview

B. VEHICLE DETECTION

Figure 3 shows the series of Envelope data when car, SUV, bus and middle-truck pass over the radar in turn, where the Envelope data change quite obviously when the vehicle passes over the radar.

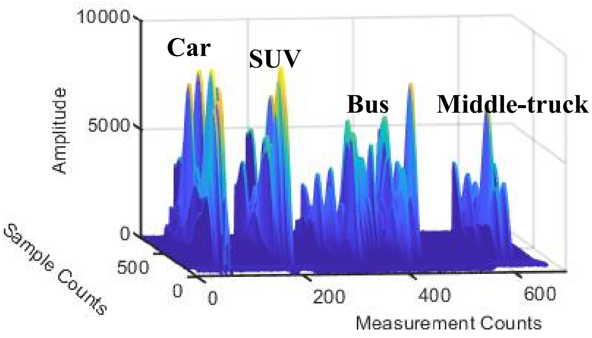


Fig. 3 Envelope data of vehicle passing over the radar

The algorithm of vehicle detection is divided into 2 steps: 1) Preliminarily divide the Envelope data into 2 categories: there is vehicle or no vehicle; 2) Filter the result by close-open operation [24].

Step 1: divide the Envelope data into 2 categories

The Envelope data generated from one measurement have quite a few samples, and we firstly fuse the samples, expressed as

$$Menv(t) = \frac{\sum_{i=1}^{n} env_i(t)}{n}, \tag{6}$$

where $Menv(t)$ is the averaged Envelope data generated from the t-th measurement.

Figure 4 shows the averaged data $Menv$ calculated from the Envelope data. The $Menv$ when the vehicle passes the radar is much larger than the $Menv$ 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

$$S(t) = \begin{cases} 0, & Menv(t) \le thre(t) \\ 1, & Menv(t) > thre(t) \end{cases} \tag{7}$$

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

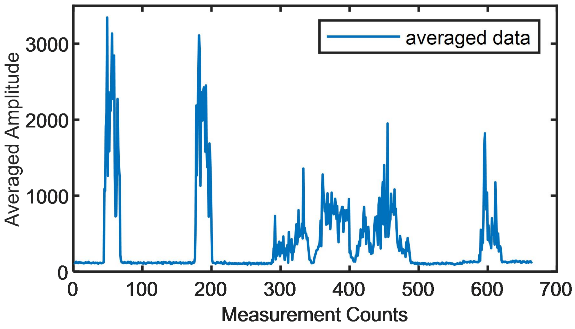


Fig. 4 Averaged data of vehicle passing over the radar

The baseline of the averaged data 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

$$thre(t) = (1 + \lambda)b(t), \tag{8}$$

where $b(t)$ is the baseline, and λ is the coefficient to adjust the threshold. The $b(t)$ is updated by

$$b(t) = \begin{cases} (1-w)b(t-1) + wb(t), S(t) = 0 \\ b(t-1), \ else \end{cases} \tag{9}$$

where w is the weighting factor to update the baseline when $S(t) = 0$.

Step 2: filtering by close-open operation

The averaged data $Menv$ 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.

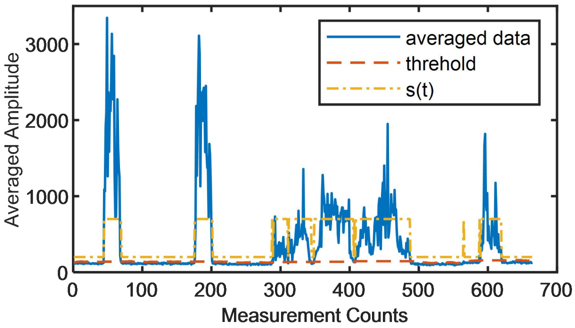


Fig. 5 Result of step 1

The method to eliminate those glitches is based on the two basic operations of Mathematical Morphology [24]: corrosion and expansion

$$(S\Theta g)(t) = \min_{m=1,2,\ldots l} \{S(t + m - l/2) - g(m)\}, \tag{10}$$

$$(S\Phi g)(t) = \max_{m=1,2,\ldots l} \{S(t + m - l/2) - g(m)\}, \tag{11}$$

where g is the structure parameter, l is the length of g, $(S\Theta g)(t)$ and $(S\Phi g)(t)$ are the results obtained by respectively corroding and expanding $S(t)$ with the structure g. In our scenario, we set

$$g(m) = 0, m = 1,2,\ldots,l, \tag{12}$$

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

$$(S\nabla g)(t) = ((S\Phi g)\Theta g)(t), \tag{13}$$

$$(S\Delta g)(t) = ((S\Theta g)\Phi g)(t), \tag{14}$$

where ∇ and Δ refer to close and open operation respectively.

The close operation can fill the gully, and the open operation can remove the spikes [24]. Therefore, we first perform the close operation on S(t) to fill the gully, then perform the open operation to remove the spikes, which is called close-open operation expressed as

$$S_{co}(t) = ((S\nabla g)\Delta g)(t), \tag{15}$$

where $S_{co}(t)$ is the filtered result of performing close-open operation on $S(t)$.

The filtered result is shown in Figure 6. Based on the filtered result, we can precisely obtain the entering and leaving measurement counts of different vehicles.

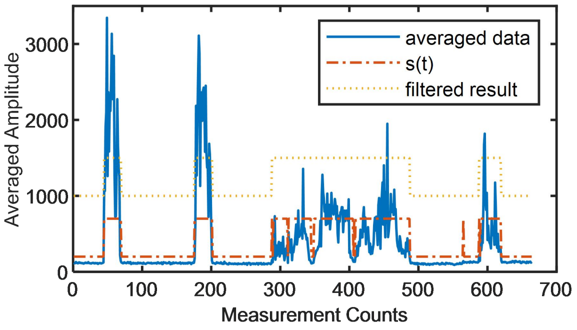


Fig. 6 Filtered result

C. RESIZE

After performing the vehicle detection, intercept the Envelope data by the entering and leaving measurement counts and each intercepted data is called a vehicle sample.

The obtained vehicle samples for car, SUV bus and middle-truck are shown in Figure 7, where the total numbers of measurement in different vehicle samples are different because of different vehicle speeds and lengths on road. The Random forest model requires the size of input data consistent, therefore we resize different vehicle samples before feature extraction and classification.

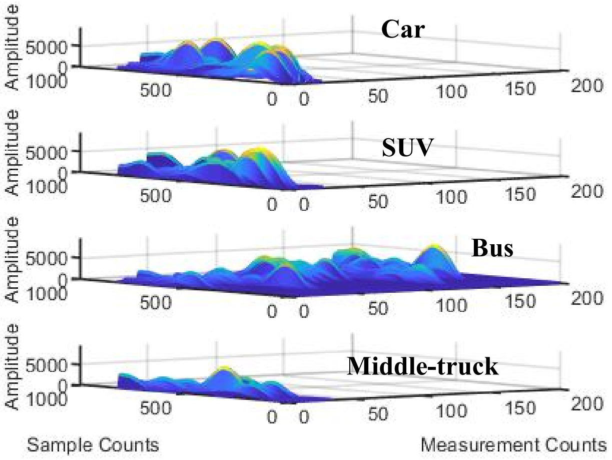


Fig. 7 Vehicle samples

We call the total number of samples as the height of vehicle sample, and the total number of measurements as the width of vehicle sample. Because the height of each vehicle sample is fixed as n, therefore we just need to adjust the width of vehicle sample.

Interpolation method can change the vehicle sample size while preserving the information as much as possible. In our scenario, we perform Linear Interpolation on each row of the vehicle sample to fix the vehicle sample size to $n * m$, where m is equal to 16. The interpolation method and the value of m were determined by experiment in Section V. Figure 8 shows the resized vehicle samples.

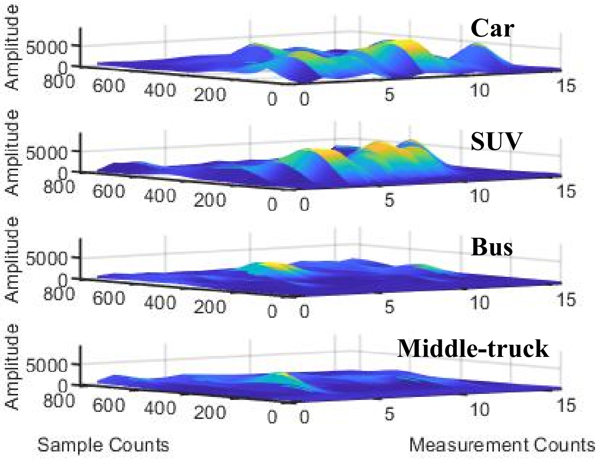


Fig. 8 Resized vehicle samples

D. FEATURE EXTRACT

Figure 9 shows the Envelope data in one resized car sample, where mc refers to measurement counts. The Envelope data generally has multiple crests because it is collected during the fast moving of vehicle, and the chassis of vehicle is uneven. In other words, the wave crest location in the Envelope data is related to the outline and height of the vehicle chassis. Therefore, the features of vehicle chassis outline and vehicle chassis height are extracted for vehicle classification.

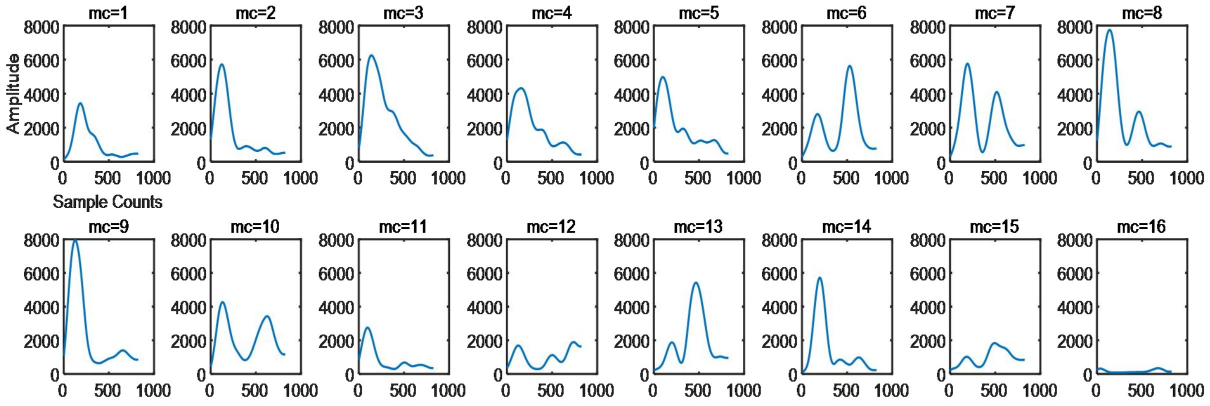


Fig.9 Each Envelope data in one resized car sample

Extraction of vehicle chassis outline

One resized vehicle sample is expressed as

$$(ENV(1),ENV(2),...,ENV(m)), \qquad (16)$$

where $ENV(j)$ is the j-th Envelope data in the resized vehicle sample. In our scenario, we extract the wave crests from every Envelope data as the feature of vehicle chassis outline, which is expressed as

$$(wcrests(1),wcrests(2),...,wcrests(m)), \qquad (17)$$

where $wcrests(j)$ is the wave crests in the j-th Envelope data.

It is a challenge to accurately calculate the wave crests of one Envelope data, because the wave crest of the Envelope data can't conform the mathematical definition of maximum points in many cases, and if we calculate the wave crest by the mathematical definition of maximum points, there will be many missed selections and multiple selections.

Therefore, we first perform the mean filter to smooth the Envelope data. Then select some candidate points from all the sample points in the smoothed Envelope data by a loose condition to avoid missing some wave crests. Finally, we select the midpoints of each region formed by candidate points as the wave crests.

Algorithm 1 shows the details for calculating the wave crests in one Envelope data, and Figure 10 shows the result of the algorithm.

Algorithm 1: Calculation of the wave crests in one Envelope data

```
$ENV = (env_1,env_i,...env_n) \Leftarrow$ One Envelope data
$CP \Leftarrow$ All candidate points in the one Envelope data
$WC \Leftarrow$ All wave crests in the one Envelope data
$ENV$ ←mean filtering on $ENV$
for $env_i$ in $ENV$ do
  if $(env_i > env_{i-1}||env_i > env_{i-2})||...||env_i > env_{i-9})$
   $\&\&(env_i < env_{i+1}||env_i < env_{i+2}||...||env_i < env_{i+9})$
      Add $(i,env_i)$ to $CP$
    end if
  end for
  for candidate point in $CP$
    if candidate point is the midpoint of the regions formed by $CP$
      Add candidate point to $WC$
    end if
  end for
return $WC$
```

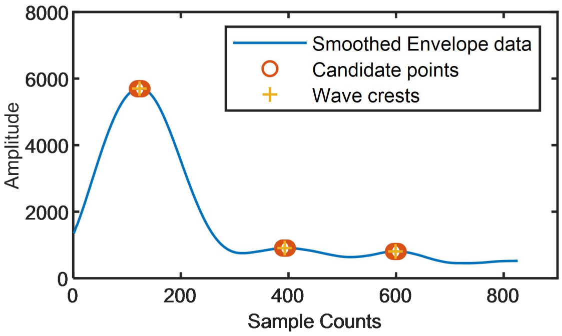


Fig. 10 Wave crests and Candidate points

Figure 11 shows the wave crest number distribution of one Envelope data in the resized vehicle sample set, where we can see only 6.31% of Envelope data has 4 or more wave crests.

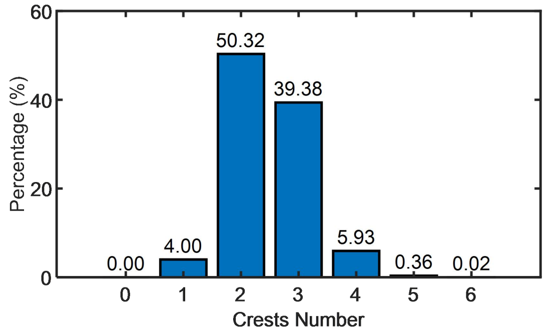


Fig. 11 Distribution of the wave crests number of one Envelope data

In order to save computing and storage resources, the number of crests extracted from the envelope data is fixed at 3. In particular, we first sort the wave crests in descending order of the wave crest height, then only keep the first 3 sets of wave crests, and fill them with 0 if there are less than 3 sets. Therefore, the wave crests in one Envelope data are fixed as the 6-dimensional vector, expressed as

$$\begin{gathered} wcrests(j) = \left(x_1^j, y_1^j, x_2^j, y_2^j, x_3^j, y_3^j\right), \\ st.\ y_1^j \geq y_2^j \geq y_3^j,\ j = 1, \ldots m, \end{gathered} \tag{18}$$

where x_i^j, y_i^j refer to the i-th wave crest in the j-th Envelope data, x_i^j is the sample counts and y_i^j is the x_i^j-th sample amplitude in the j-th Envelope data.

Extraction of vehicle chassis height

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 firstly approximately calculate the height based on each piece of Envelope data in the resized vehicle sample

$$h_j = x_1^j \Phi_{res} + d_{start}, j = 1, \ldots, m \tag{19}$$

where h_j is the height calculated by the j-th Envelope data in the resized vehicle sample, x_1^j is the sample counts of the highest wave crest in the j-th Envelope data according to Equation 18. Then average the heights

$$vh = \frac{\sum_{j=1}^{m} h_j}{m}, \tag{20}$$

where vh is the feature of vehicle chassis height extracted from the vehicle sample.

E. RANDOM FOREST MODEL

After the feature extraction, we could obtain the features of vehicle chassis outline and height, which are expressed as a feature vector

$$fv = (wcrests(1), \ldots, wcrests(m), vh), \tag{21}$$

where fv is the feature vector of size $m * 6 + 1$. In the last step of the proposed approach, we categorize the feature vector into vehicle type based on the machine learning algorithm Random forest [25], 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 20 decision trees, and the number is determined by the experiment in Section V. Each decision tree is trained in turn by a subset of the feature vector set. For the decision tree splitting, the best feature is selected from the K features, which are randomly selected from the unselected features, and K is set as $\sqrt{m * 6 + 1}$, which is the sqrt of the feature vector size. Each decision tree stops splitting until it cannot be split. After training, we obtain the Random forest model, which divides the road vehicle into 4 types: car, SUV, bus and middle-truck.

V. EXPERIMENTS

A. 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 d_{start} and d_{length} to 10 cm and 40 cm. According to Equation 2, the dimension of Envelope data generated from one measurement is 826.

The road vehicle has different length and speed. If the measurement frequency of PCR is too low, it's unable to detect the vehicle of 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.

The PCR working in Envelope mode filters each Envelope data by an exponential smoothing filter, which reduce the response of Envelope data when vehicle passed over the radar. Therefore, we set the weight (average-fact) of the filter as 0 to forbidden it.

Tab. 1 Experimental parameters

Parameter	Description	Value
d_{start}/m	the closest distance that PCR can detect	0.1
d_{length}/m	the length of the distance interval that PCR can detect	0.4
n	the dimension of one Envelope data	826
f/HZ	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. However, even a motorcycle passed by the PCR at a very close distance, the Envelope data still has no response, therefore it's difficult to distinguish motorcycles. For that, our classification task doesn't include distinguishing motorcycles.

With the configurations of Table 1, we collect data on multiple roads in Dongguan, China. The one scene of collecting data is shown as Figure 12. The detection node integrated with the radar PCR is deployed in the center of the lane, and the gateway and the host computer are placed near the detection node and connected through a serial port, and a mobile phone is used to record the vehicle model. The gateway receives the data of the detection node, and the host computer saves it locally. Finally, 1,281 vehicle data are obtained, including 315 cars, 324 SUVs, 342 buses, and 300 middle-trucks.

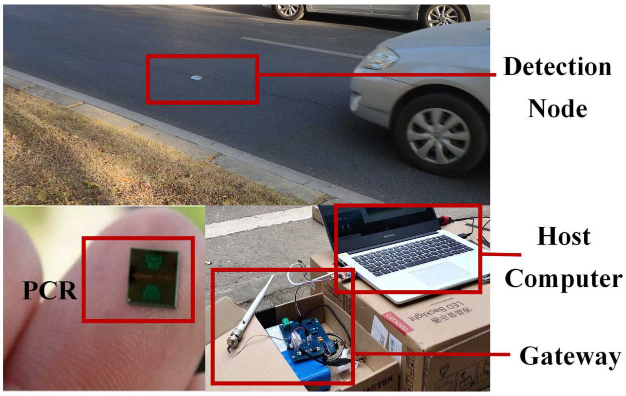


Fig. 12 Experimental scenario

B. SELECTION OF VEHICLE DETECTION PARAMETERS

In this section, we configure the parameters λ, w and l in Equations 8, 9 and 12 for the best performance of the vehicle detection algorithm, and the results are concluded in Table 2.

Tab. 2 Configuration of vehicle detection parameters

Parameter	Description	Value
w	the weight to update the baseline	0.2
λ	the coefficient to adjust the threshold	0.2
l	the length of close-open operation	17

As shown in Figure 4, the averaged data $Menv$ change acutely and sometimes fluctuate below the threshold especially when a bus passed over the radar. Therefore, there are some missed judgments based on the method shown in Equation 7. In order to avoid the baseline being incorrectly stretched by these data of missed judgments, 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 λ and l are determined by the actual data. To confirm the best values of λ and l, we set different λ and l to calculate the accuracy of vehicle detection on the whole collected data, and the result is shown in Figure 13, where we conclude λ 0.2 and l 17 are the best configurations. In addition, there is a correlation between λ and l, which is that when λ is bigger and the l should be bigger too to get good performance in general. Because when λ is bigger, the threshold becomes bigger and there are less incorrect but more missed judgments, which causing wider gully in $S(t)$, then the l should be bigger to fill the wider gully.

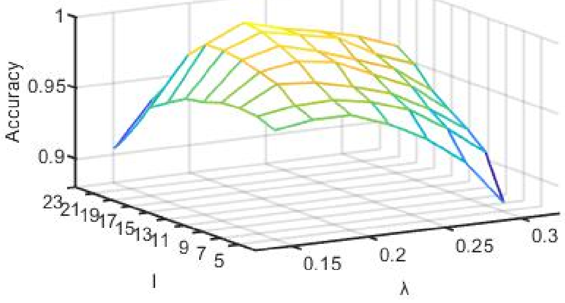


Fig. 13 Accuracy of vehicle detection with different l and λ

C. SELECTION OF INTERPOLATION METHODS

AND RESIZED VEHICLE SAMPLE WIDTH

The interpolation method and the resized vehicle sample width m will affect the performance of vehicle classification. Figure 14 shows the average accuracy of vehicle classification with different interpolation methods including Nearest Neighbor Interpolation, Linear Interpolation and Cubic Interpolation [26], where m is between 2 and 100.

When m is between 8 and 22, Cubic Interpolation and Linear Interpolation have better accuracy than Nearest Neighbor Interpolation. However, when m is greater than 22, the accuracy of Cubic Interpolation is reduced, while the accuracy of Linear interpolation is relatively stable, and is always greater than the accuracy of Nearest Neighbor Interpolation. Therefore, we choose Linear Interpolation to resize the vehicle sample. In addition, the larger m is, the more storage and computing resources are consumed, and when m is greater than 16, the accuracy of linear interpolation changes slowly, therefore m is determined to be 16.

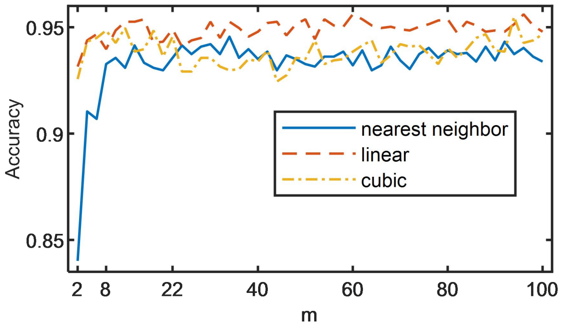


Fig. 14 Accuracy of vehicle classification with different m and interpolation methods

D. SELECTION OF TREES NUMBER IN RANDOM FOREST MODEL

The number of decision trees is a key parameter that determines the accuracy, running speed and storage cost of the Random Forest model. Figure 15 shows the out-of-bag error [25] with different numbers of decision trees in Random Forest model. The error doesn't change much after the number of decision trees is greater than 20, therefore we configure the number of decision trees in Random Forest model as 20.

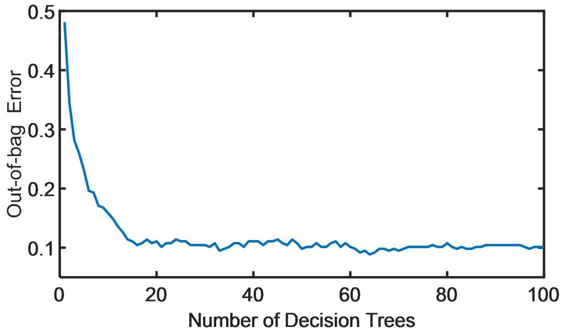


Fig. 15 Out-of-bag error with different number of decision trees

E. COMPARISON ALGORITHM AND PERFORMANCE INDICATORS

The approach proposed in this article is called VCRF. Another method is implemented for comparison experiments:

VCSVM: the vehicle classification algorithm based on Support Vector Machine (SVM) [27]. 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.

The 5-fold cross-validation method [28] is used to evaluate the two 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.

For a type of vehicles, we define the following concepts to calculate the performance indicators.

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

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

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

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

Then the performance indicators accuracy, precision and recall can be calculated by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{22}$$

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

F. EXPERIMENTAL RESULTS

The detail classification results with the two algorithms are summarized in Table 3 and 4, respectively.

It can be seen from the Table 3 that car and SUV have lower accuracy, precision and recall compared with bus and middle-truck, which indicates that there are more incorrectly judgments between car and SUV, that's because the chassis of car and SUV is more similar and more difficult to distinguish.

The bus has the highest accuracy, 99.30%. This is because the chassis of the bus is very different from other types of vehicles, furthermore the chassis of different buses are also very similar because the bus model in a city is relatively fixed.

It can be seen from the Table 4 that the comparison algorithm VCSVM has a bit lower accuracy than VCRF.

Tab. 3 Classification results with VCRF

	Car	SUV	Bus	Middle-Truck
Accuracy	89.93%	88.99%	99.30%	97.89%
Precision	74.60%	88.61%	99.12%	91.74%
Recall	89.52%	64.81%	98.25%	100%

Tab. 4 Classification results with VCSVM

	Car	SUV	Bus	Middle-Truck
Accuracy	88.52%	87.82%	98.59%	96.96%
Precision	73.73%	81.82%	98.21%	89.91%
Recall	82.86%	66.67%	96.49%	98.00%

Figure 16 shows the time required for the algorithms VCRF and VCSVM to classify different numbers of vehicle samples, where VCRF requires more time. Table 5 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.

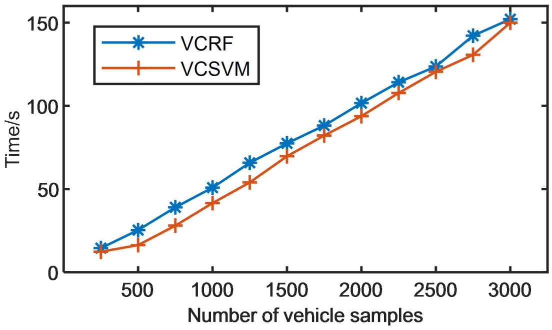


Fig. 16 Classification efficiency comparison

Tab. 5 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 extract the features of vehicle chassis outline and height from the intercepted vehicle data, which is used as the input of a Random Forest model. The model is trained by the features set calculated by all the intercepted vehicle data, and categorizes the vehicle into 4 types: car, SUV, bus and middle-truck. The experimental result has shown the averaging accuracy of our approach is 94.02%.

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 [29].

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] W. Chen, Z. Zhang, X. Wu, and J. Deng, "On-Road vehicle detection algorithm based on Mathematical Morphology," in *Proc. WASA*, Sep. 2020, pp. 11-19.

[9] R. Fu, M. Zhang and C. Wang, "Behavior analysis of distant vehicles using LIDAR point cloud," *Cluster Computing*, vol. 22, no. 2, pp. 8613-8622, Jul. 2019.

[10] Eum, Bae, Jeon, Lee and Oh, "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, May 2017.

[11] C. Jang, C. Kim, K. Jo and M. Sunwoo, "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, Feb. 2017.

[12] A. Acconeer. "Radar sensor introduction," Sep. 2020, [Online]. Available: 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] W. Li, Z. Liu, Y. Hui, L. Yang, R. Chen and X. Xiao, "Vehicle classification and speed Estimation based on a single magnetic sensor," *IEEE Access*, vol. 8, pp. 126814-126824, 2020.

[15] S. Kaewkamnerd, J. Chinrungrueng, R. Pongthornseri and S. Dumnin. "Vehicle classification based on magnetic sensor signal," in *Proc. IEEE ICIA*, Jun. 2010, pp. 935-939.

[16] R. Girshick, "Fast R-CNN," in *Proc. IEEE ICCV*, 2015, pp. 1440-1448.

[17] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: towards real-time object detection with region proposal networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "SSD: single shot multiBox detector," in *Proc. ECCV*, Oct. 2016, pp. 21-37.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. IEEE CVPR*, Jun. 2016, pp. 779-788.

[20] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proc. IEEE CVPR*, Jul. 2017, pp. 6517-6525.

[21] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[22] W. Lyu, Q. Lin, L. Guo, C. Wang, Z. Yang and W. Xu, "Vehicle detection based on an improved faster R-CNN method: regular section," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* vol. 104, no. 2, pp. 587-590, Feb. 2021.

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

[24] Y. Bai, S. Yu and J. Li, "A new generalized open and close Morphological filters," *Journal of Image and Graphics*, vol. 14, no. 8, pp. 1523−1529, Aug. 2009.

[25] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[26] Y. You and X. Zhou, "Research of optimal interpolation algorithm for digital image," *Chinese Space Science and Technology*, no. 3, pp. 14-18, Jun. 2005.

[27] X. Zhang, "Statistical learning theory and support vect

or machines," *Acta Automatica Sinica*, vol. 26, no. 1, pp. 32-41, Jan. 2000.

[28] L. Yang and Y. Wang, "A review of various cross-validation estimation methods for generalization errors," Application Research of Computers, vol. 32, no. 5, pp. 1287-1290, May 2015.

[29] X. Zhang, Z. Zhang, Y. Li, X. Guo and W. Chen, "V ehicle detection algorithm based on pulse coherent radar," *Computer Engineering and Applications*, early access, Mar. 2021, doi: 11.2127.TP.20210331.1502.014.