A Common Traffic Object Recognition Method Based on Roadside LiDAR

1st YuLin Xu
School of Electronic and Optical Engineering
Nanjing University of Science and Technology
Nanjing, China
lxxuyulin@163.com

2nd Wei Liu
Three Fast Online
Beijing Science and Technology Co
Beijing, China
kk.ii@163.com

3rd Yong Qi
School of Intellectual Property
Nanjing University of Science and Technology
Nanjing, China
790815561@qq.com

4th YiHan Hu
School of Electronic and Optical Engineering
Nanjing University of Science and Technology
Nanjing, China
1585687088@qq.com

5rd Weibin Zhang*
School of Electronic and Optical Engineering
Nanjing University of Science and Technology
Nanjing, China
weibin.zhang@njust.edu.cn

2022 IEEE 7th International Conference on Intelligent Transportation Engineering (ICITE) | 978-1-6654-6007-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICITE56321.2022.10101469

***Abstract*—Environment perception and object recognition are the key to realize cooperative vehicle infrastructure system. This paper proposes an object recognition method based on roadside LiDAR, which realizes pedestrian, bicycle and vehicle recognition through feature extraction and convolutional neural networks. First, the DBSCAN clustering algorithm is used to segment the point clouds from LiDAR. In order to decrease space complexity, the traditional features are optimized by feature selection. Then, a lightweight VGGNet network is built as the recognition network, and the self-attention mechanism is added to improve the representation ability of features and recognition accuracy. Experimental results show that the proposed method has a recognition rate of 87% for common traffic objects, which has good robustness and real-time performance.**

***Keywords*—3D object detection, point clouds, convolutional neural network, LiDAR**

I. Introduction

The environment perception system is a key component of the future cooperative vehicle infrastructure system, which is also the information source for autonomous vehicles to make driving decisions. Real-time object detection is an important task of the environment perception system and is the key to achieve cooperative vehicle infrastructure system.

LiDAR is a sensor that can provide accurate 3D information by sensing the environment in all directions. It can obtain the obstacle distance and reflection intensity by transmitting multiple laser beams and calculating the time interval of reflected laser beams, and then generate point clouds data. Lidar is widely used in autonomous driving because of its large amount of data, high precision and strong adaptability to the environment [1].

Obstacle detection and recognition algorithms based on LiDAR point clouds data can be divided into two categories, one is based on manual feature extraction and classification, and the other is an end-to-end method based on deep learning networks, which are different in data feature extraction [2]. The traditional manual feature extraction algorithm maps the original point clouds data into the feature space through 3D point clouds operators, while the deep learning algorithm automatically learns the feature through the network. The traditional detection algorithm can detect objects in real-time, but the recognition accuracy needs to be further improved in complex scenarios. The deep learning recognition algorithm depends on the training data set. Although it has good effect on the fixed type of LiDAR and the corresponding installation position, it doesn't have high generalization ability. Once the conditions change, the model needs to be retrained with new data set.

In the traditional detection algorithm, objects are detected and recognized by background filtering, point clouds segmentation, object clustering and object recognition. Local features or global features are extracted and machine learning methods, such as support vector machines and clustering [3], are used to classify point clouds objects. P. Ghamisi and B. Höfle [4] used the aggregated local point neighborhoods to directly extract features from the LiDAR point clouds and proposed a composite kernel support vector machine method for classification. Results indicated that this method can obtain high classification accuracy with LiDAR data alone. Reference [5] shows how point clouds data from a 16-beam LiDAR sensor are processed to extract useful information and features to classify semi-trailer trucks hauling ten different types of trailers.

With the development of artificial intelligence, deep learning algorithms have made a breakthrough in object recognition. However, due to the irregularity of point clouds data, typical deep learning recognition algorithms for 2D images cannot be directly used for 3D point clouds data. Scholars have come up with three approaches to solve this problem. The first one is to

This work is supported by National Nature Science Foundation under Grant 71971116.

project 3D stereo point clouds onto multiple 2D planes and extract view features from them, then fuse these features for classification. MVCNN [6] is a pioneering network of this approach, which simply max-pools multi-view features into a global descriptor. However, max-pooling only retains the maximum elements from a specific view, resulting in information loss. The second one is to voxelize the point clouds into 3D grids and format the irregular point clouds data into a neat matrix for convolution filtering. For example, VoxNet [7] applies a three-dimensional convolutional neural network (3D CNN) for shape classification of point clouds. The third one extracts features from point clouds data directly. For example, PointNet [8] ingeniously uses symmetric functions to solve the disorder problem of point clouds and introduces a T-Net network to learn the rotation of point clouds, making the model more adaptable to the translation and rotation of point clouds. However, PointNet [8] cannot capture the local structural information of the point clouds and cannot to process details. Based on this, the author improved the network and proposed PointNet++ [9], which realizes point clouds segmentation and local feature extraction by designing a local domain sampling representation method and a multi-level encoding-decoding structure.

Based on PointNet [8], scholars put forward some point clouds recognition networks. PointRCNN [10] uses PointNet++ [9] to segment point clouds into foreground points and background points, then extracts features from the foreground points for recognition. Point-GNN [11] is a graph neural network for 3D object detection. The input point clouds are encoded as the nearest neighbor graph with a fixed radius, and then the graph is put into Point-GNN for recognition.

In practical applications, deep learning networks cannot meet the real-time requirements due to the complex computation, so manual feature extraction is still used in many cases. Aiming at the problem of pedestrian detection and tracking of autonomous vehicles using a LiDAR, Reference [12] used statistical features and a classifier trained by Support Vector Machine (SVM) to recognize pedestrians, and improved the recognition performance with the aid of tracking results. For transportation objects detection based on roadside LiDAR, Reference [13] used clustering to identify traffic objects and optimized the unsupervised clustering algorithm to solve the problem of over-segmentation, so as to meet the real-time requirements while maintaining high clustering accuracy.

In this paper, a traffic objects recognition method based on roadside LiDAR is proposed. By performing background filtering, point clouds segmentation, object clustering and feature extraction, then using a lightweight VGGNet network [14] to classify the objects, realizing the recognition of three common traffic objects, vehicles, bicycles and pedestrians. The contribution of this paper includes the following three aspects. Firstly, in order to reduce the space complexity of the system, the dimension of viewpoint feature histogram (VFH) is reduced by feature selection. The second is to synthesize geometric features, statistical features and filtered VFH features, and put forward a composite feature vector as the input of the neural network to retain as much information as possible. Thirdly, simplify the VGGNet structure and add the self-attention mechanism to improve the correlation between features, so as to improve the classification accuracy.

The rest of this paper is organized as follows. In section II and III, the system architecture is introduced, and the proposed algorithm is explained in detail. Section IV describes the experiments and results, and we conclude the paper in section V.

II. System Architecture

The overall architecture of the system is composed of four parts, as shown in Fig. 1. The first step is data preprocessing which includes downsampling, filtering, and ground segmentation. Then the point clouds are segmented by a clustering algorithm, as can be seen in Fig. 1(b). Thirdly, features of the clustering results are extracted. Finally, the recognition results are obtained by the neural network.

A. Point Cloud Preprocessing

In order to reduce computation load, point clouds are downsampled and regions of interest are selected according to geometric features for further processing.

In addition, the existence of ground points will affect the recognition of objects, so the ground points should be filtered out. We use the random sampling consensus (RANSAC) algorithm to fit the ground. RANSAC algorithm can divide the point clouds into inliers that can be used to represent the model and outliers that are not applicable to the model. For a plane model, the points within a certain distance from the model are inliers, and the model with the most inliers is optimal. RANSAC algorithm returns the optimal model through multiple iterations, and the inliers represent the ground point clouds that will be filtered.

B. Clustering Algorithm

After filtering out the background point clouds, the remain point clouds belong to different categories, which may be traffic objects such as pedestrians, bicycles, and cars, or interference objects such as trees, traffic signs, traffic lights. The point clouds of these objects are not only different in spatial distribution, but also uncertain in quantity, so clustering is needed to realize the segmentation of point clouds. Due to the irregular distribution of LiDAR point clouds, we adopt the density-based DBSCAN algorithm for clustering.

The DBSCAN algorithm defines the cluster as the maximum set of points that are densely connected. It can cluster objects with arbitrary shapes in noisy spatial datasets according to the density of samples. There are two important parameters in the DBSCAN algorithm, ε is the neighborhood radius when defining the density, and $minpts$ is the threshold of elements in the neighborhood radius of an element. The point where the number of points within the neighborhood radius ε reaches $minpts$ is called the core point.

When classifying LiDAR point clouds clusters, first select a point p from the data arbitrarily. If p is the core point for parameter ε and $minpts$, find all points whose density are reachable from p to form a cluster, which is the target. If the selected point p is not the core point, select another point until it is the core point. Repeat above steps until all points are processed, then the object point clouds can be segmented by clustering.

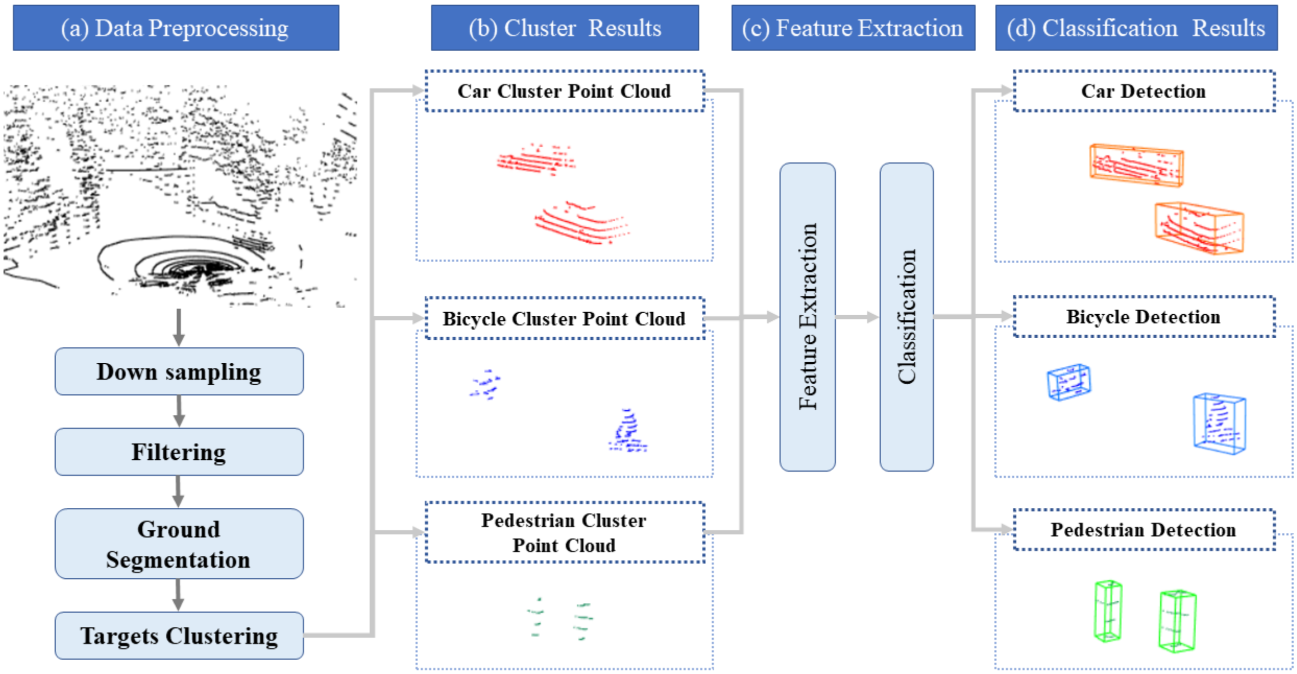

Fig. 1. The proposed system architecture. The main process are divided into four parts: (a)data processing, (b)clustering, (c)feature extraction, (d)Classfication.

C. Feature Extraction

LiDAR point clouds have rich feature information. However, the object size varies due to the distance between the object and LiDAR changes, and sometimes the object shape is incomplete because of occlusion. Hence, we integrate geometric information, statistical information and viewpoint feature histogram (VFH) of the LiDAR point clouds, and use a composite feature vector to describe the object.

Geometric information mainly refers to the length, width, height, volume and aspect ratio of the object.

Statistics information represents the overall distribution of the object. For a point clouds object, its covariance matrix can be calculated by (1).

$$\sum = \begin{bmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{coz}(z,y) & \mathrm{cov}(z,z) \end{bmatrix} \tag{1}$$

Three eigenvalues can be obtained by solving covariance matrix, as shown in (2).

$$\lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} (\lambda_1 > \lambda_2 > \lambda_3) \tag{2}$$

After sorting by size, the eigenvalues can represent the overall distribution of the point clouds. If one eigenvalue is particularly larger, it indicates that the distribution of the point clouds is close to linear distribution. If there are two large eigenvalues, it means that the distribution of the point clouds is close to plane distribution. If there is little difference between the three eigenvalues, it indicates that the distribution of the object is roughly equal in all directions, which is similar to a spherical uniform distribution.

Therefore, by combining these three eigenvalues, the linear coefficient, planar coefficient, divergent coefficient, and curvature of the point clouds can be obtained to describe the overall distribution of the point clouds. Table I lists the calculation formulas of the four statistical features and some examples of values corresponding to traffic objects.

TABLE I. STATISTICAL FEATURE FORMULAS AND SOME EXAMPLES

Statistical Features	Formulas	Pedestrian	Bicycle	Vehicle
Linear Coefficient	$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$	0.937	0.3372	0.873
Planar Coefficient	$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$	0.026	0.539	0.076
Divergent Coefficient	$S_\lambda = \frac{\lambda_3}{\lambda_1}$	0.036	0.087	0.042
Curvature	$C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$	0.032	0.050	0.000072

The viewpoint feature histogram (VFH) describes the global features of point clouds objects, which consists of an extended fast point feature histogram (FPFH) that describes the surface shape and a component between the viewpoint direction and the normal. The default VFH descriptor consists of two parts. The first one is three angle components and one distance component generated by the FPFH descriptor, each component uses 45 binning subdivisions, a total of 180 binning subdivisions. And the other is the viewpoint component value histogram with 128 binning subdivisions. So VFH has a total of 308 dimensional features. Fig. 2 shows the VFH examples of the vehicle, bicycle and pedestrian.

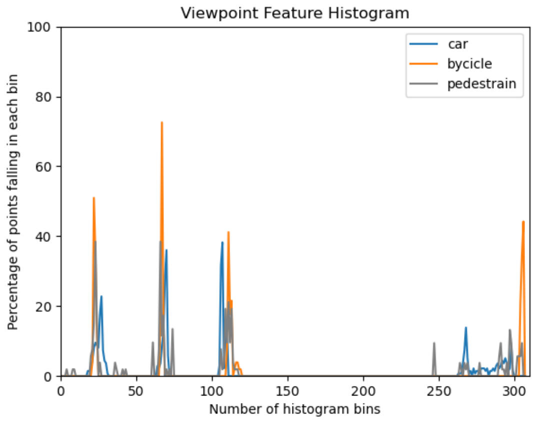

Fig. 2. The VFH of vehicle, bicycle and pedestrian

As can be seen from the figure, VFH values of different objects are quite different and can be used as classification features. Besides, since many values in the VFH graph are zero, we select the useful components of VFH features to reduce space complexity.

We choose the chi-square test to filter VFH features. The chi-square test is an independence test for discrete variables. If a feature is independent of the label, it means that this feature is useless for prediction. Therefore, the chi-square test is an important method to eliminate irrelevant features. By calculating the chi-square statistics between features and labels, the top 100 features with the highest scores are retained, as shown in Fig. 3. It can be seen that the selected 100 features retain most of the structure of the original features.

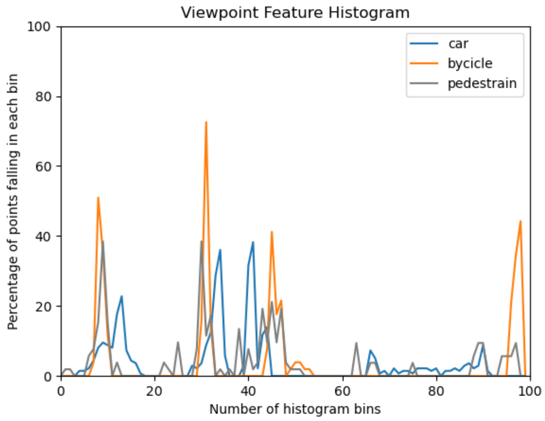

Fig. 3. The selected VFH of vehicle, bicycle and pedestrian

III. Object Recognition

A. VGGNet

VGGNet is an effective classification network which consists of 5 convolution layers and 3 fully connected layers. The layers are separated by max-pooling. The activation units of hidden layers are the ReLU function. The block structure of VGGNet is shown in Fig. 4.

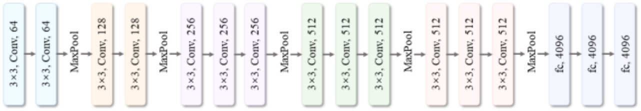

Fig. 4. The structure of VGGNet

B. Self-Attention Mechanism

The self-attention mechanism performs attention calculation on the input sequence to obtain important information by assigning weights to every element.

The attention mechanism regards the input features as key-value pairs <*Key*, *Value*>. Given an element *Query* in the target, by calculating the correlation between the *Query* and each *Key*, the weight coefficient of the *Value* is obtained. Then add the weighted sum of *Value* to obtain the attention value. So essentially, the attention mechanism is a weighted sum of the input elements, while *Query* and *Key* are used to calculate the weight coefficient of the corresponding *Value*. Usually use *Q*, *K*, *V* to represent *Query*, *Key*, *Value* respectively. The general definition of the attention mechanism is shown as (3).

$$\text{Attention}\,(Q,K,V)=soft\max\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{3}$$

where $Q\in R^{n\times d_k}, K\in R^{m\times d_k}, V\in R^{m\times d_v}$, d_k is the scaling factor, which represents the dimensions of *K*. If d_k is large, it will cause a large dot product, which will push the softmax function to the region with minimal gradients. To counteract this effect, $\frac{1}{\sqrt{d_k}}$ is used to scale dot product.

For the self-attention mechanism, the *Q*, *K*, *V* come from the same data source, that is to say, *Q*, *K*, *V* are obtained by different linear changes of the input feature matrix. The overall structure of the self-attention mechanism is shown in Fig. 5.

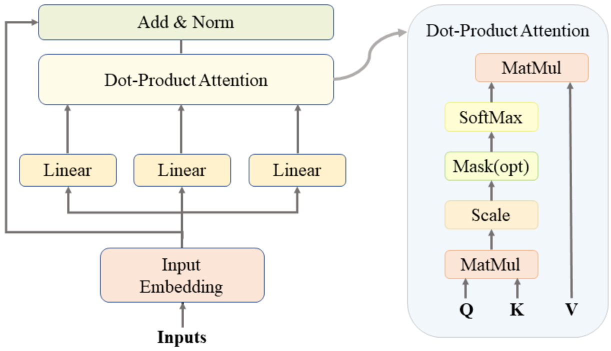

Fig. 5. The structure of self-attention machanism

C. SA-VGGNet

We build a lightweight VGGNet as the classification network and add the self-attention mechanism to the network.

Since the correlation between each feature and target is different, we introduce the self-attention mechanism to improve the representation ability of composite features, so that the neural network can pay more attention to outstanding features during training and recognition, so as to improve the performance of the recognition network.

The network structure is shown in Fig. 6. The extracted features are fed into the self-attention mechanism layer, and the high-dimensional features are obtained by convolution layers, then they are input into full connection layers for classification.

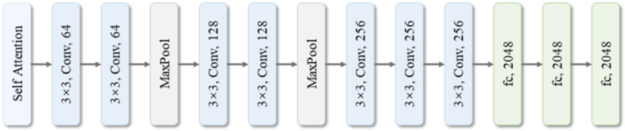

Fig. 6. The structure of SA-VGGNet

IV. Experimental Results and Analysis

In this section, the quantitative evaluation is carried out to verify the effectiveness of the traffic objects recognition algorithm proposed in this paper. We use the robosense RS-LiDAR-16 to obtain the point clouds data. We set up the LiDAR on the roadside of the main road on campus, and collect three typical traffic objects, pedestrians, bicycles and vehicles. We processed 30,000 frames of data and obtained 22121 objects. Compared with the actual recorded data, the clustering accuracy is 89.3%. The clustering accuracy of various objects is shown in Table Ⅱ.

TABLE II. Clustering Accuracy of Objects

Description	Vehicle	Bicycle	Pedestrian	Total
Actual Object Number	4659	6405	13707	24771
Clustering Object Number	4268	5777	12076	22121
Detection Rate	91.9%	90.2%	88.1%	89.3%

The 22121 objects obtained by clustering were divided into training data set and testing data set according to the ratio of 8:2. The data for training and testing is shown in Table Ⅲ.

TABLE III. Objects Number of Training Set and Testiing Set

Description	Vehicle	Bicycle	Pedestrian
Training Set	3368	4673	9663
Testing Set	900	1104	2413

In training, the training data set is divided into training set and validation set according to the ratio of 8:2, and the model with the highest accuracy of the validation set is found by adjusting parameters. The loss of the training set and the accuracy of the validation set are calculated. The training loss curve and validation accuracy curve of the model is shown in Fig. 7. The solid line represents the training loss and validation accuracy of the proposed SA-VGGNet, and the dashed line represents the training result of the lightweight VGGNet only. It can be seen that the model converges faster after adding the self-attention mechanism, and the validation accuracy is also improved.

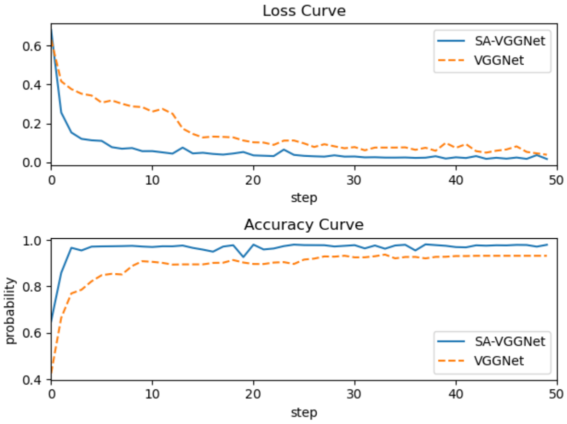

Fig. 7. The loss curve and accuracy curve of SA-VGGNet and VGGNet

Fig. 8 shows the confusion matrix obtained from experiments on the testing set with the SA-VGGNet. It can be seen that the SA-VGGNet performs well in vehicle recognition. However, there is a little false detection between pedestrians and bicycles, which may be because the target was blocked at that time, resulting in information loss. The overall recognition rate of clustering results is 97.5%, and the comparison results between the proposed method and other methods are shown in Table IV.

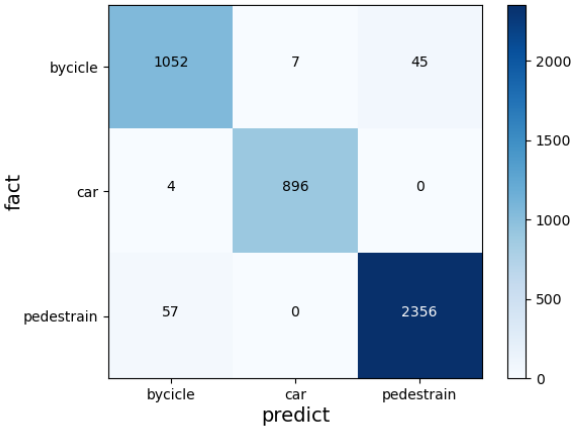

Fig. 8. Confusion matrix of the testing set

TABLE IV. Comparison Results With Other Methods

Accuracy	Vehicle	Bicycle	Pedestrian	Total
SVM	89.2%	83.3%	88.4%	88.0%
Random Forest	86.8%	85.1%	85.3%	85.6%
Proposed Method	99.5%	95.2%	97.6%	97.5%

V. Conclusion

In order to accurately recognize pedestrians, vehicles and bicycles in urban transportation environments, this paper proposes a traffic object recognition method based on roadside LiDAR. The point clouds collected by a 16-beam lidar are preprocessed and segmented by the DBSCAN clustering algorithm. Then the geometric feature, statistical feature and viewpoint feature histogram are calculated as fusion features. The self-attention mechanism and convolutional neural networks are used to classify objects. The final recognition accuracy is 87%.

In future research, we will further improve the recognition accuracy and rate by improving recognition algorithms. We will use the current results to obtain object trajectory data, and use tracking algorithms, such as Kalman filtering, to reduce the missed detection rate.

Acknowledgment

This work was supported by the National key research and development program: Key projects of international scientific and technological innovation cooperation between governments (Grant 2019YFE0123800), and National Natural Science Foundation of China (Grant 71971116).

References

[1] Y. Li and J. Ibanez-Guzman, "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems," in IEEE Signal Processing Magazine, vol. 37, no. 4, pp. 50-61, July 2020.

[2] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 12, pp. 4338-4364, 1 Dec. 2021, doi: 10.1109/TPAMI.2020.3005434.

[3] F. Gao, C. Li and B. Zhang, "A Dynamic Clustering Algorithm for LiDAR Obstacle Detection of Autonomous Driving System," in IEEE Sensors Journal, vol. 21, no. 22, pp. 25922-25930, 15 Nov.15, 2021, doi: 10.1109/JSEN.2021.3118365.

[4] P. Ghamisi and B. Höfle, "LiDAR Data Classification Using Extinction Profiles and a Composite Kernel Support Vector Machine," in IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 5, pp. 659-663, May 2017.

[5] Olcay Sahin, Reza Vatani Nezafat & Mecit Cetin (2022) Methods for classification of truck trailers using side-fire light detection and ranging (LiDAR) Data, Journal of Intelligent Transportation Systems, 26:1, 1-13.

[6] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multiview convolutional neural networks for 3D shape recognition,”in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 945–953.

[7] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4490–4499.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 77–85.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5105–5114.

[10] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 770–779.

[11] W. Shi and R. Rajkumar, “Point-GNN: Graph neural network for 3D object detection in a point cloud,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 1711–1719.

[12] Heng Wang, Bin Wang, Bingbing Liu, Xiaoli Meng, Guanghong Yang,Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle,Robotics and Autonomous Systems,Volume 88,2017,Pages 71-78,ISSN 0921-8890.

[13] S. Kim, J. Ha and K. Jo, "Semantic Point Cloud-Based Adaptive Multiple Object Detection and Tracking for Autonomous Vehicles," in IEEE Access, vol. 9, pp. 157550-157562, 2021.

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, May 2015.