Research on Classification of Rod-Shaped Ground Objects in Road Scene Based on Multi-Feature Associated Network

Pan Xiang, Li Yongqiang, Li Jiajia, Yang Junli, Zang Jing

School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, Henan, 454000, China

Abstract: *Vehicular LiDAR technology provides powerful technical support for the accurate acquisition of rod-shaped ground objects' spatial information in road scenes. However, how to solve the accurate extraction and classification of rod-shaped ground objects is always a technical problem to be solved. Aiming at the above problems, this paper proposes a MFA-Net based on improved PointNet Network. Firstly, PointNet is used to extract point features and global features from the input data. Then, the local information of the point cloud is extracted by the joint module of high and low layer features constructed by the upsampling method, and the point features, global features and local features are fused into joint features. Finally, the accurate classification of point clouds is achieved by the fully connected layer. Experimental analysis shows that the proposed algorithm model has achieved good results on the public datasets ModelNet40 and ModelNet10, and the classification accuracy on the self-made rod-like ground object dataset Rod5 is 99.0%, which verifies the excellent classification performance and robustness of the proposed algorithm model.*

Keywords: *PointNet; Local feature; Graph convolution; Feature combination; Rod-shaped ground objects; Vehicle-borne LiDAR*

1. Introduction

Pole features (street lamp, street tree, pole, sign, etc.) are important components of road scenes, and how to efficiently and automatically extract and classify pole features is a current research hotspot, which is of great significance for the construction of smart roads and smart cities[1]. As a cutting-edge mapping technology[2], vehicle-borne LiDAR can quickly obtain comprehensive 3D spatial information of various features in the road scene, and the dense point cloud can accurately express the 3D structure information of rod-shaped features in the road scene, which provides powerful data support for deep learning to solve the rod-shaped feature classification problem. The current research on point cloud classification based on deep learning mainly includes.

(1) Based on the multi-view method, representative algorithm models such as MVCNN, GVCNN, Auto-MVCNN, etc[3-8]. This kind of algorithm firstly converts the point cloud into multi-angle projection maps, then uses the two-dimensional image processing method to analyze these projection maps containing the point cloud information, and finally realizes the classification by combining the multi-view features. This kind of algorithm has high efficiency, but the method of reducing 3D to 2D results in the loss of part of the geometric information of the point cloud, and can only obtain the features from a specific perspective, which is difficult to satisfy the automatic classification of large-scale complex scene features.

(2) Voxel-based method, representative algorithm models such as VoxNet, 3D ShapeNes, OctNet, etc[9-14]. This kind of algorithm transforms unstructured point clouds into regular structures with voxels as units, and then uses convolutional neural networks such as 3DCNN for feature learning to complete classification. This kind of algorithm can effectively solve the problem of unstructured point cloud, but the low resolution voxel will lose useful information, and the high resolution will lead to greater computing pressure.

(3) Based on the graph convolution method, representative algorithm models such as GCN, DGCNN, RGCNN, etc[15-20]. This kind of algorithm combines graph structure with convolution operation and obtains feature information through the relation between graph nodes. This kind of algorithm can effectively obtain the geometric structure information of point cloud, but due to the limitation of the

number of fully connected network layers, too many parameters are prone to overfitting problems.

(4) Point based method, representative algorithm models such as PointNet, Pointnet ++, So-Net, etc[21-28]. This kind of algorithm directly processes the original data to extract features, and can make full use of the information of the original point cloud. However, its focus is the extraction of global features, and the study of local features is still not deep enough. PointNet, as a pioneering algorithm, deals directly with point clouds to greatly facilitate the point cloud processing process, but further research on the deeper level of point cloud information is still needed.

Although deep learning has become one of the effective methods to solve the classification problem, the current research on point cloud classification based on deep learning still has the following problems: (1) the network models are mostly used in existing publicly synthesized datasets, with less application in actual field point cloud data and insufficient generalization capability; (2) the ability to capture point cloud information still has room for in-depth research. Therefore, this paper makes use of the home-made pole feature dataset of road scene point clouds acquired by vehicle-borne LiDAR, and applies the actual collected data in the deep learning network. And considering that PointNet can only extract global features, we propose to build a Multiple Feature Associated Network (MFA-Net) to further enhance the point cloud feature extraction capability, and finally achieve the classification of vehicle-borne LiDAR rod-shaped feature point clouds.

2. Principle of the algorithm

To further improve the accuracy of deep learning in solving rod feature classification, this paper proposes a multi-feature joint network model, as shown in Figure 1. The model firstly uses PointNet to extract point features and global features from the input data, while the local features of the point cloud are extracted through the joint module of high and low level features constructed by up-sampling method, and then the point features, global features and local features are fused into joint features, and finally the point cloud is accurately classified through the fully connected layer.

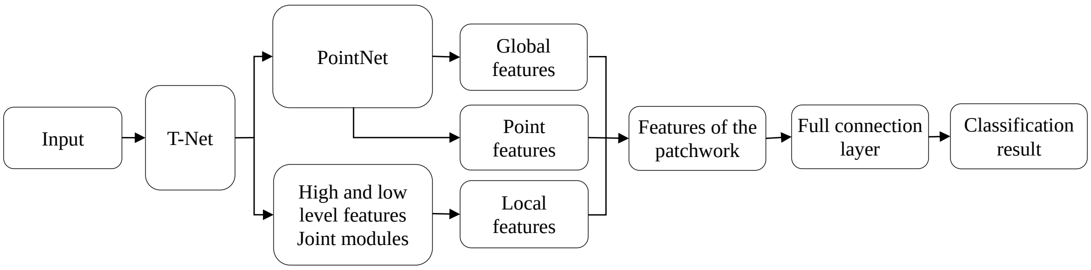


Figure 1: MFA-Net network model

2.1 PointNet

PointNet is a deep neural network model that directly processes unordered point cloud data, and the formal representation is shown in formula (1).

$$f(x_1, x_2, \cdots, x_n) \approx g[h(x_1), h(x_2), \cdots, h(x_n)], x_i \in R^D \tag{1}$$

where: $x_1, x_2, \dots, x_n$ denotes the input unordered point cloud data; $x_i \in R^D$, R denotes real number and D denotes dimension; the number of points is n; the ith point cloud is denoted as x_i; g represents the Maxpooling symmetry function; function h represents the multilayer perceptron (MLP); f denotes the continuous set function that maps a set of points to a vector.

Figure 2 shows the PointNet network structure, where the input is the 3D coordinates of N points (N × 3), and the alignment of the input data is achieved by a 3D spatial transformation matrix prediction network T-Net (3). The aligned data are processed through a shared parameter MLP (64, 64) model to extract the 64-dimensional features of each point. The features are then aligned by acting on the features of each point via T-Net (64) to predict a 64 × 64 transformation matrix. Then a three-layer perceptron MLP (64, 128, 1024) is used to project the feature-based points into the high-dimensional space for feature learning, directly changing the dimensionality of the features to 1024. finally, the global features of the point cloud (1×1024) are extracted through the maximum pooling layer (Maxpooling), and the global features are processed using the fully connected layer FC(512,128,k). The scores of the K categories are output to achieve the classification of the point cloud.

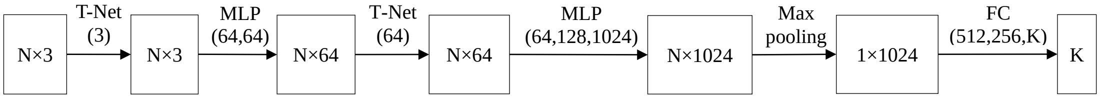


Figure 2: PointNet network structure

2.2 Point feature and global feature extraction

In PointNet networks, after the original data is processed by T-Net, a low-dimensional convolution operation is first performed, and then convolution is performed by the multilayer perceptron MLP to directly obtain the 1024-dimensional global features. This approach, although it can describe the global information of the points, will lose the detailed information of the points. In order to extract richer point cloud information, in this paper, N×64 is extracted as features of the points in the process of extracting global features. To further prevent information loss, the features of (1×64) points are saved by aggregation through a maximum pooling operation. Finally, the point features, global features and local features extracted by the joint module of high and low level features are spliced and fused, and the joint features are processed using the fully connected layer to output the classification results, and the extraction process is shown in Figure 3.

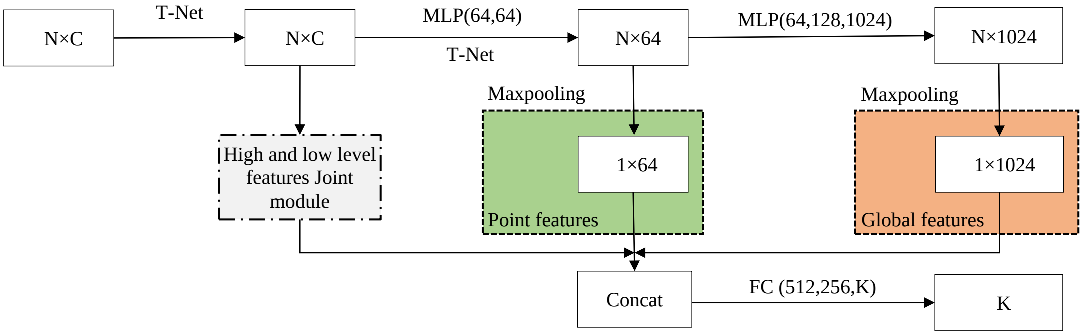


Figure 3: Flow chart of point featureand global feature extraction

2.3 Construction of joint modules for high and low level features

Only the features of each individual point are considered in the PointNet classification network, which does not take into account the local information of the point cloud. In view of this, this paper extracts local feature information by constructing a high and low level feature joint module, the structure of which is shown in Figure 4. The module consists of the SA module, the GraphConv module and the Unite module, the function of the SA module is to obtain the local features of the point cloud, and subsequently the enhanced intermediate level features are obtained step by step through the GraphConv module, in order to enrich the semantic information of the local features with the Unite module. The specific implementation steps are as follows: the input N×3 matrix is aligned by T-Net to obtain the N×3 matrix (where N represents the number of points and the subsequent different N values represent different number of point clouds). There are two subsequent paths for N×3: First, the N×128 low-level features were obtained by GraphConv module. Second, the N_1×256 middle-layer features were obtained through the SA module and GraphConv module, and then the results of the second SA module were obtained through the GraphConv module to get N_2×512 high-level features. High-level features, middle-level features, and low-level features obtain features with richer semantic information through Unite module. Finally, combine the high-level features with the features obtained by the two Unite module modules to obtain the local features. By concatenating point features, global features and local features, this paper uses the full connection layer to process the combined features and output the classification results.

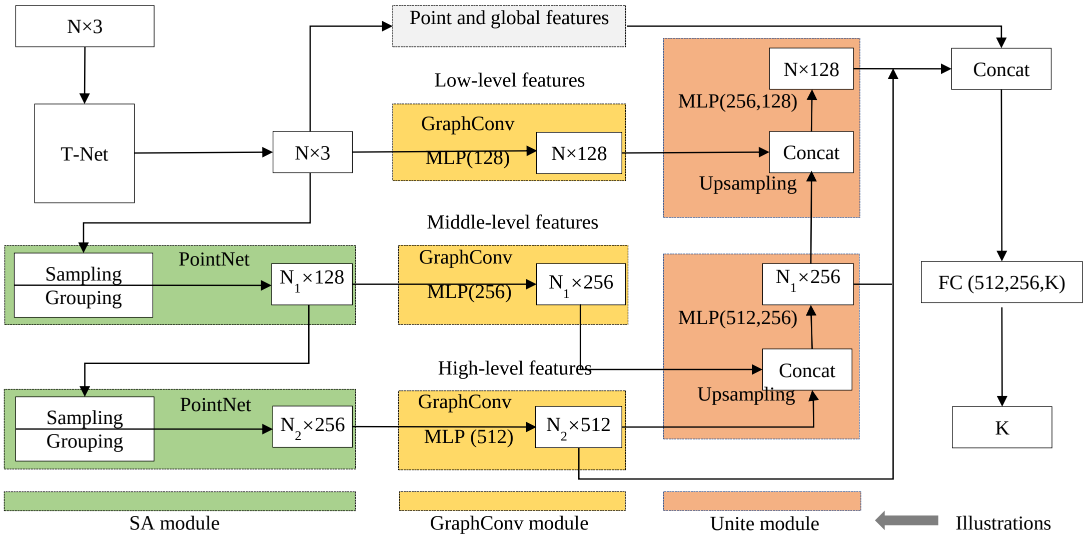


Figure 4: Structure diagram of high and low layer feature joint module

2.3.1 SA module

In this paper, the local features of points are obtained through the SA module, which is divided into three parts: sampling, grouping and feature extraction (PointNet), as shown in Figure 5. A certain number of points are first selected using Sampling; then a specified number of point sets are clustered by grouping with each sampled point as the centre; finally the point sets are iteratively processed using PointNet to aggregate the features to each sampled point. The results of the first SA module are fed into the next SA module to repeat the operation, extracting fewer and fewer centroids, but containing more and more feature information, thus obtaining the local features of the points. Using the first SA in Figure 4 as an example. The input number of points N is downsampled to N_1, with each sampling point as the center, and the nearest K points with a specified number are clustered. Then, the input N×3 matrix was aggregated to obtain $N_1 \times K \times 3$ matrix. Finally, local features in the sampling area were extracted by PointNet to obtain $N_1 \times 128$ results.

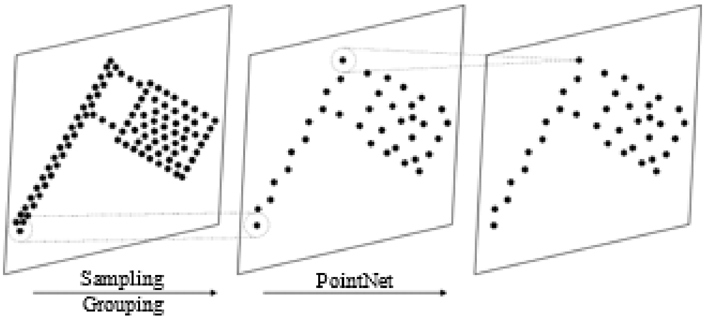


Figure 5: Flow chart of SA module

In this paper, farthest point sampling (FPS) and K nearest neighbor (KNN) algorithm are used respectively. FPS algorithm process: Firstly, a random point in the original data is taken as the initial point, and then the point nearest euclidean distance from the initial point is selected from the remaining point set to become the second point in the sampling group, and the iterative operation meets the sampling quantity requirement[29].The FPS algorithm can reduce the number of points and the overall geometry of the point cloud without changing the overall geometry of the point cloud, which can reduce the computational complexity of the network model. KNN algorithm process: Every point after sampling is taken as the central point, euclidean distance between the remaining points and the central point is calculated, and K points nearest to each central point are selected to form the region group of each central point. For each central point, the closer the point is to the central point, the stronger the connection. Therefore, this algorithm can effectively aggregate the required points in the region and strengthen the ability of local information aggregation. PointNet's task is to aggregate the features of regional point groups, add them to each central point feature, and ultimately achieve local feature extraction of points.

2.3.2 GraphConv module

The original point cloud is fed into the two-layer SA module for processing by means of layer advancement, and finally the next higher level features of the point cloud are obtained. The number of points contained in the next higher level features is relatively small, but the semantic information contained is richer. In contrast, the lower-level features in the input layer contain more points that are closer to the original point cloud, with a larger number of points, but contain less semantic information. Therefore, this paper proposes a layer-by-layer approach to obtain the enhanced features of the intermediate layers through the GraphConv module for subsequent operation of the Unite module. The basic steps are as follows: before the original data is spatially transformed and input to the first SA module, the N×128 low-level features are obtained by GraphConv module. Similarly, the output of the first SA is passed through GraphConv module to obtain N_1×256 middle-level features, and the output of the second SA is passed through GraphConv module to obtain N_2×512 high-level features. In this paper, the Unite module is used to combine the high level-middle level-low level features in order to enrich the semantic information, and finally the combined features are fed to the classification network.

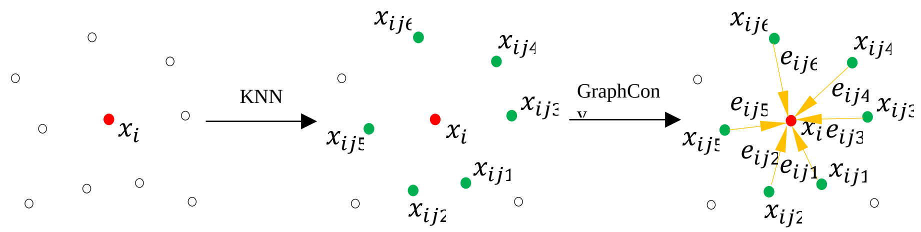


Figure 6: Feature extraction of GraphConv (k=6)

The GraphConv module realizes feature aggregation through graph convolution operation, so as to enhance the ability of feature capture. The graph convolution constructs a directed graph G=(V, E) by the KNN algorithm, where V={1,...,N} represents the vertices of the graph structure, N is the number of point clouds, and the edges formed by KNN are denoted by E. Taking K=6 as an example, Figure 6 shows the graph convolution feature extraction process. For a selected node x_i , the KNN algorithm is used to select the nearest K nearest neighbours as $\{x_{ij1}, x_{ij2},\dots, x_{ijk}\}$, the distance between node x_i and its nearest neighbours as the edges of the graph structure, e_{ij} denotes the edge features generated by node x_i and its nearest neighbours, and the feature aggregation of all nodes' directed edges is the output result of graph convolution, which is formally expressed as:

$$x_i = \sum_{j:(i,j)\in 1,\dots,k} h_\Theta(x_j - x_i) \tag{2}$$

where $h_\Theta(x_j - x_i)$ denotes the edge function e_{ij} considering only edge features, where h_Θ is a series of non-linear functions parameterised by the set of learnable parameters Θ, capable of achieving $R^D \times R^D \to R^{D'}$ feature learning.

2.3.3 Unite module

The purpose of Unite module is to attach the features of the previous layer with fewer points and rich semantic information to the features of the current layer through the method of feature up-sampling, so as to enrich the semantic information of the current layer. Figure 7 shows the structure of the Unite module. As can be seen from Figure 7, the upper layer features are first transformed by upsampling, the results are spliced onto the current layer features, and finally the fused features are output by MLP. The new features have two paths: one is directly to the last step; the other is as input to the next layer of the Unite module.

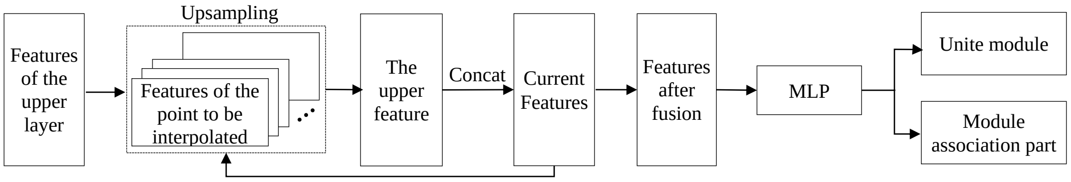


Figure 7: Unite module structure diagram

Through up-sampling, the lower layer features are transferred to the upper layer, which can make the lower layer features get richer semantic information. Upsampling in this module is implemented by the reverse interpolation method. Euclidean distance matrix and weighting coefficient are calculated by the points of two adjacent layers. Three nearest points are selected near each point to be interpolated, and weighted average values are calculated according to the characteristics of the three points as the features of the interpolation points. The features of the interpolation points are stacked with those of the upper layer by means of jump connection to realize the upsampling of features. Where, the weighting coefficient is obtained by dividing the reciprocal distance of each point by the sum of the reciprocal distance of the three points. The number of point clouds in the feature of the lower layer is restored to the number of point clouds contained in the feature of the upper layer by interpolation method, and the features are connected into fusion features. The feature difference is calculated as shown in Formula (3).

$$\hat{f}_i = \frac{\sum_{j=1}^{M} \omega_j(p_i) f_j}{\sum_{j=1}^{M} \omega_j(p_i)} \quad ,\text{where}$$

$$\omega_j(p_i) = \begin{cases} \dfrac{1}{\|p_i - p_j\|_2}, & p_j \in N(p_i) \\ 0, & otherwise \end{cases} \tag{3}$$

where: $\hat{f}_i$ represents the eigendifference of the points to be interpolated, where p_i refers to the known points, p_j refers to the unknown points, and f_j represents the eigenvalue information of the known points; $\omega_j(p_i)$ represents the weight value, which is the inverse of the euclidean distance between the unknown and known points, where $N(p_i)$ represents the set of known point cloud regions.

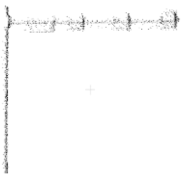
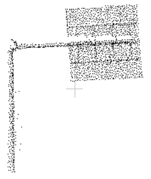
3. Experimental analysis

3.1 Experimental Environment

The experimental hardware environment is Inter Core i7-9700F + RTX2060(6G) + 16GB RAM, and the software environment is Ubuntu16.04 ×64 + Windows10 ×64 + CUDA10.1 + cuDNN7.5 + TensorFlow1.13 + Python3.7.

3.2 Experimental data

Table 1: Sample effect of Rod5 dataset

Name/Label	**Street lamp /0**	**Traffic light /1**	**Street tree /2**	**Pole /3**	**Traffic sign /4**
Point cloud					

(1) Open dataset. Two subsets of ModelNet[10], ModelNet40 and ModelNet10, was chosen for the experiments. The public dataset called ModelnNet10 contains 12311 rigid 3D models of 40 different classes, of which 9843 are training models and 2468 are testing models, including cars, chairs, doors and so on. The public dataset called ModelnNet10 contains 4900 3D CAD models from 10 different categories, of which 3991 are training models and 909 are testing models, including bathtubs, beds, sofas and so on.

(2) Rod5 dataset. The SSW-3 vehicle-borne LiDAR system was used to collect the point cloud data of the ring road around Henan Polytechnic University. The non-ground point cloud was obtained through pre-processing such as denoising, cropping, chunking and cloth filtering, and then the extraction of rod features was realised based on the binary image segmentation algorithm, followed by sampling, quantity augmentation, normalisation and label annotation, etc. Finally, h5 file was written in 7:3 ratio to complete the production of rod-shaped ground object dataset (This dataset is named Rod5 in this article). Sample visualization is shown in Table 1. The Rod5 dataset contains 3277 pole feature models in 5 categories, including 2355 training models and 922 test models, including street lamp, traffic light, street tree, pole

and traffic sign.

3.3 Precision analysis of open data set

In order to verify the effect of MFA-Net classification network in this paper, ModelNet40 and ModelNet10 public datasets were used for testing, and the accuracy was compared with PointNet and PointNet++ networks. In the classification network of this paper, the values of FPS and KNN algorithm of the first SA module are selected as 512 and 32 respectively, and the values of FPS and KNN algorithm of the second SA module are selected as 128 and 64 respectively. The GraphConv module selects 40, 30 and 20 values of the graph convolution of low-level, middle-level and high-level features respectively according to the feature level. According to the experimental environment of this paper, linear rectifier ReLU is used for the activation function, and cross entropy is used for the loss function. ADAM optimizer is used to guide the deep learning network to update the parameters, and momentum gradient descent method is used for training. After each fully connected layer, dropout with the parameter of 0.5 is added. Other parameters are set as follows: batch_size=8, decay_rate=0.7, momentum=0.9, learning_rate=0.001, max_epoch=251, num_point=2048. The algorithm was evaluated from the aspect of classification accuracy through the test set, and the specific values were shown in Table 2.

As can be seen from the analysis of Table 2 below, the accuracy of the improved MFA-Net classification network on ModelNet40 and ModelNet10 is 91.0% and 94.2%, respectively, achieving good results, which are 2.4% and 2.6% higher than the PointNet network, and 1.2% and 1.9% higher than the PointNet++ network. The reason is that PointNet can only obtain the global features of point cloud, and Pointnet ++ can only extract the local features of point cloud. However, the MFA-Net network in this paper extracts the local features through the high-low layer feature association module, and combines the point features and global features, so the point cloud information extraction ability is stronger.

Table 2: Comparison of classification accuracy of each algorithm

Network	ModelNet40	ModelNet10
PointNet	88.6%	91.6%
PointNet++	89.8%	92.3%
MFA-Net	91.0%	94.2%

3.4 Experimental analysis of the Rod5 dataset

The aim of this paper is to study deep learning to solve the classification problem of typical features in road scenes, so the following experiments are trained and tested using the rod feature dataset, and the four evaluation metrics of overall accuracy, average accuracy, iterative performance and robustness are selected to compare the MFA-Net network with other methods.

3.4.1 Precision analysis

The network parameters were set as in Section 2.3. As can be seen from Table 3, the overall classification accuracy of the improved MFA-Net classification network for pole features was 99.0%, including 100% for street lamp, 96.2% for traffic light, 100% for street tree, 100% for pole and 98.8% for traffic sign. The analysis shows that street tree and poles are more homogeneous and different in shape compared to the other three features, so the classification accuracy is higher. Traffic light had the lowest classification accuracy due to their greater variety, containing cameras, pedestrian crossing signals, intersection signals and road centre signals with varying shapes.

In order to verify the classification effect of pole-shaped features, PointNet, PointNet++ and MFA-Net in this paper were used to compare the accuracy. From Table 3, it can be seen that the MFA-Net classification network in this paper has the highest overall classification accuracy of 99.0% for rod-shaped features, which is 2.8% higher than the PointNet network and 0.8% higher than the PointNet++ network. From the perspective of single sample classification accuracy, the accuracy of MFA-Net network classification in this paper is all higher than or equal to PointNet and Pointnet ++. The average classification accuracy is the average of all sample independent accuracies, and as can be seen from the table, the MFA-Net network has the highest average classification accuracy of 99.0%.

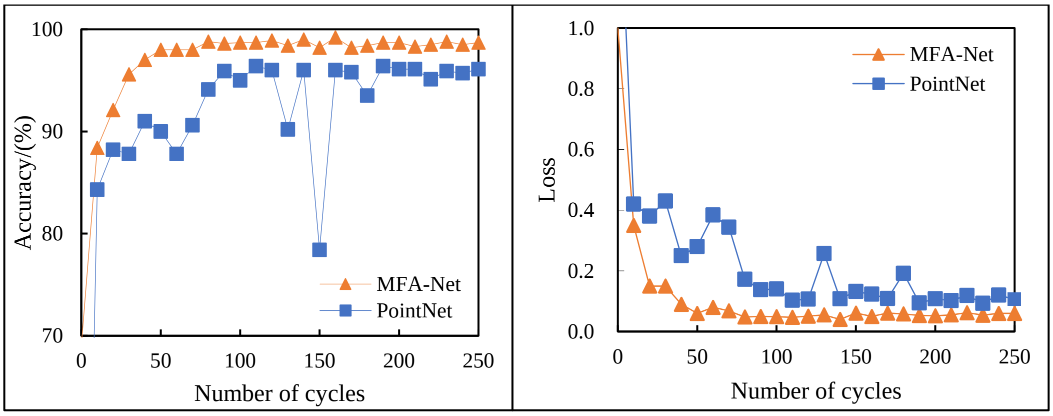
Through analysis, PointNet classification network can only learn the global features of points, and PointNet++ can only learn local features, and the ability to capture point cloud feature information is insufficient. On the basis of extracting global features from PointNet, the MFA-Net classification network in this paper adds a high and low level feature association module for extracting local features, so as to acquire features between points at a deeper level. At the same time, combined with point features, further enhance the feature extraction ability of the network, with a high classification accuracy. Experiments show that the improved MFA-Net classification network has better performance.

Table 3: Precision analysis of Rod5 dataset

Category	PointNet	PointNet++	MFA-Net
Street lamp	98.1%	99.1%	100%
Traffic light	94.3%	93.7%	96.2%
Street tree	100%	100%	100%
Pole	99.4%	98.7%	100%
Traffic sign	87.4%	98.8%	98.8%
Overall accuracy	96.2%	98.2%	99.0%
Average accuracy	95.8%	98.1%	99.0%

3.4.2 Iteration performance analysis

In order to more fully analyze the classification performance of the MFA-Net network in this paper, the Rod5 dataset is taken as the test object, and the accuracy change curve and loss change curve of the dataset along with the number of iterations are constructed, as shown in Figure 8. As can be seen from Figure 8 (a), MFA-Net network convergence speed is faster than PointNet, and classification accuracy is always higher than PointNet. With the increase of the number of iterations, the accuracy of the PointNet network will fluctuate greatly, while the MFA-Net network is relatively stable. As can be seen from Figure 8 (b), compared with PointNet network, MFA-Net network has a faster and relatively stable loss decline, and the loss function value is smaller when the network converges, which indicates that MFA-Net network has a greater fitting ability to the test data set. This is mainly due to the design of high and low level feature association module in MFA-Net network, which captures the local features of the point cloud in a deeper level, combines the point features with the global features, and further strengthens the feature extraction ability. By synthesizing Figure 8 (a) and (b), it can be seen that the classification capability of MFA-Net network is stronger than that of PointNet network, which fully reveals the superiority of MFA-Net network in this paper.



(a) Precision training curve *(b) Loss training curve*

Figure 8: Training curves of PointNet and MFA-Net networks

3.4.3 Robustness analysis

For the deep learning of point cloud, the number of point clouds has a significant impact on the characteristics. Generally speaking, the denser the point cloud, the more accurate the extracted features will be, while the sparser the point cloud, the more sketchy the extracted features will be. Therefore, this paper compares the robustness of MFA-Net and PointNet network through sparsity experiment. Under the condition that other experimental parameters were consistent, the number of point clouds was reduced from 2048 points to 1024, 512 and 256 points for training and testing. The robustness of MFA-Net and PointNet in this paper was compared through input sparse point learning, and the precision comparison was shown in Figure 9. As can be seen from Figure 9, the accuracy of the PointNet network attenuates from the highest 96.2% to 95.1%, while the accuracy of MFA-Net network in this paper is basically above 98.0% by combining point features, global features and local features, and the accuracy of the improved MFA-Net network is always higher than that of the PointNet network. In the case of sparse point clouds, it can still have higher accuracy and show better robustness.

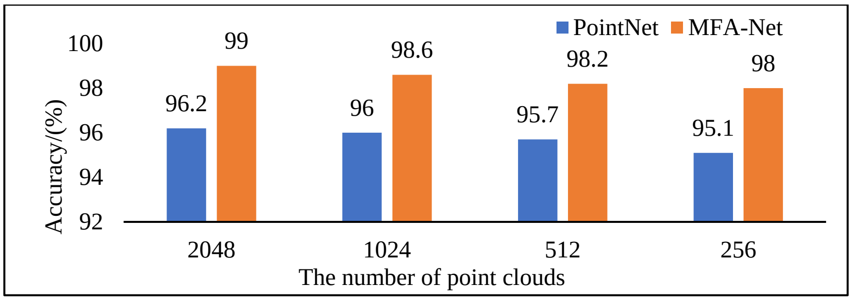


Figure 9: Precision comparison of different sampling points

4. Conclusion

In this paper, an improved deep learning network model MFA-Net is proposed to effectively solve the classification problem of rod-shaped ground objects in road scenes. Rod-shaped ground objects were extracted from the point cloud processed by vehicle LiDAR and made into the dataset needed for research. Considering that PointNet failed to take into account the local features of point, a high-low feature association module was designed to extract the local features of point cloud. By combining point feature, global feature and local feature, the feature extraction capability of point cloud is further strengthened. The experiment shows that MFA-Net network has better performance on open dataset. The classification accuracy of rod-shaped ground objects is 99.0% when the rod-shaped ground objects data set is put into MFA-Net network, which is 2.8% higher than that of PointNet network, proving that the improved deep learning network model is effective in solving rod-shaped ground objects in road scenes.

References

[1] Li Y Q, Li P P, Dong Y H, et al. Automatic Extraction and Classification of Pole-Like Objects from Vehicle Lidar Point Cloud [J]. Acta Geodaetica et Cartographica Sinica, 2020, 49(06):724-735.
[2] Lu X S, Liu R F, Tian M Y, et al. The Improved Mathematical Morphology Method is Used to Carry out Vehicular Laser Point Cloud Ground Filtering[J]. Geomatics and Information Science of Wuhan University, 2014, 39(05):514-519.
[3] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3d shape recognition[C].Proceedings of the IEEE International Conference on Computer Vision, 2015: 945-953.
[4] Qi C R, Su H, Niessner M, et al. Volumetric and multi-view cnns for object classification on 3d data [C].Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 5648-5656.
[5] Cao Z, Huang Q, Karthik R. 3D object classification via spherical projections[C]//2017 International Conference on 3D Vision (3DV). IEEE, 2017: 566-574.
[6] Feng Y, Zhang Z, Zhao X, et al. GVCNN: group-view convolutional neural networks for 3d shape recognition[C].Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 264-272.
[7] Yu T, Meng J, Yuan J. Multi-view harmonized bilinear network for 3d object recognition [C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018:186-194.
[8] Li Z, Wang H, Li J. Auto-MVCNN: neural architecture search for multi-view 3d shape recognition [J]. 2020, Arxiv Preprint Arxiv: 2012. 05493.
[9] Maturana D, Scherer S. Voxnet: a 3d convolutional neural network for real-time object recognition[C]. 2015 IEEE/ RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 922-928.
[10] Wu Z, Song S, Khosla A, et al, 3D shapeNets: a deep representation for volumetric shapes[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015:1912-1920.
[11] Huang J, You S. Point cloud labeling using 3d convolutional neural network[C]//2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 2016: 2670-2675.
[12] Riegler G, Osman Ulusoy A, Geiger A. Octnet: learning deep 3d representations at high resolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 3577-3586.
[13] Hua B S, Tran M K, Yeung S K. Pointwise convolutional neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 984-993.

[14] Le T, Duan Y. Pointgrid: A deep network for 3d shape understanding[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9204-9214.

[15] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J]. Arxiv Preprint Arxiv: 1609.02907, 2016.

[16] Wang C, Samari B, Siddiqi K. Local spectral graph convolution for point set feature learning[C]. // Proceedings of the European Conference on Computer Vision (ECCV). 2018: 52-66.

[17] Te G, Hu W, Zheng A, et al. Rgcnn: regularized graph cnn for point cloud segmentation [C]. Proceedings of the 26th ACM International Conference on Multimedia, 2018: 746-754.

[18] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds [J]. ACM Transactions on Graphics (TOG), 2019, 38(5): 1-12.

[19] Liang Z, Yang M, Deng L, et al. Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds[C].2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 8152-8158.

[20] Lu Q, Chen C, Xie W, et al. PointNGCNN: deep convolutional networks on 3d point clouds with neighborhood graph filters [J]. Computers & Graphics, 2020, 86: 42-51.

[21] Qi C R, Su H, Mo K, et al. Pointnet: deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 652-660.

[22] Qi C R, Yi L, Su H, et al. Pointnet++: deep hierarchical feature learning on point sets in a metric space [J]. Advances in Neural Information Processing Systems, 2017, 30.

[23] Li J, Chen B M, Hee Lee G. So-net: self-organizing network for point cloud analysis [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9397-9406.

[24] Li Y, Bu R, Sun M, et al. PointCNN: convolution on x-transformed points[C]. Advances in Neural Information Processing Systems, 2018:820-830.

[25] Bai J, Xu H. MSP-Net: Multi-Scale Point Cloud Classification Network [J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(11):1917-1924.

[26] Liu Y, Fan B, Xiang S, et al. Relation-Shape convolutional neural network for point cloud analysis [C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019:8895-8904.

[27] Li R, Li X, Heng P, et al. Point Augment: an auto-augmentation framework for point cloud classification[C].Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020:6377-6386.

[28] Yang B S, Han X, Dong Z. A Deep Learning Network for Semantic Labeling of Large-Scale Urban Point Clouds [J]. Acta Geodaetica et Cartographica Sinica, 2021, 50(08):1059-1067.

[29] Liang Z M, Zhai Z L, Zhou W. 3D Point Clouds Classific-ation Based on Multi -Scale Dynamic Graph Convolution Network. [J]. Computer Applicati-ons and Software, 2021, 38(05):263-267+306.