Classification method of LiDAR point cloud based on three-dimensional convolutional neural network

Zhongyang Zhao¹, Yinglei Cheng¹*, Xiaosong Shi¹ and Xianxiang Qin¹

¹ Information and Navigation College, Air Force Engineering University, Xi’an, Shanxi, 710077, China

*Corresponding author’s e-mail: Ylcheng718@163.com

Abstract. Aiming at the classification problem of ground objects in complex scenes of airborne LiDAR data, this paper proposes an algorithm based on three-dimensional convolutional neural network (3D-CNN). The algorithm improves the pre-processing method of LiDAR point cloud and realize automatically classification of LiDAR point cloud in complex scenes. Based on the input of 3D-CNN is voxel grid, this paper selects multi-scales to construct voxel grids for each point in the point cloud, trains them with the network, and then combines the features of multi-scales through the fully connected layer. Finally, the network returns the category score of each point to complete the classification of LiDAR point cloud. The algorithm is verified by the Vaihingen dataset provided by ISPRS. The experimental results show that the proposed algorithm can achieve higher classification accuracy than other convolutional neural networks dealing with point clouds.

1. Introduction
At present, LiDAR point cloud data classification methods mainly include feature extraction and using deep neural networks. The method based on feature extraction mainly uses the elevation, intensity, echo times and other spatial context information of the point cloud to extract features and design the classifier for classification. In the existing methods, methods based on feature extraction are widely used [1-3]. However, such methods require some prior knowledge and need to verify the validity of the extracted features.

The method of deep learning has made remarkable achievements in image processing. However, due to the disorder, unstructured and dimension of point cloud data, deep learning needs further research in point cloud data processing. Most existing studies convert point cloud data into image or voxel mesh forms before feature learning for use in neural networks. In [4, 5], the point cloud is transformed into voxel grid, and a three-dimensional convolutional neural network is applied to the voxel grid for point cloud classification and segmentation. However, the voxel grid representation will cause information loss and affect the classification accuracy due to the sparsity of the point cloud data. Literature [6,7] projects 3D point clouds into 2D images and uses 2D convolutional neural networks for classification, but it is limited by large-scale scene classification and semantic understanding.

In this paper, the point cloud processing method is improved based on the voxel grid to represent the point cloud. Instead of voxel meshing all point clouds, a size voxel grid is built for each point in the point cloud. The voxel grids of different scales are selected as the input of three-dimensional convolutional neural network to complete classification of LiDAR point cloud data.
2. Algorithm
Due to the disorder and unstructured point cloud, voxel meshing is required as an input to 3D-CNN. However, the voxel grid loses point cloud information and reduces the accuracy of the classification. In this paper, the point cloud processing is improved, so that the point cloud can be classified into 3D-CNN without losing the point cloud information. The algorithm is different from voxel meshing for all point clouds in point cloud processing, but constructs a voxel grid of size \( n \times n \times n \) for each point in the point cloud. Different voxel grid sizes are selected as different scales. The voxel grids of different scales are input into 3D-CNN to extract features. Finally, the fully connected layers are used to combine features of different scales to complete the classification of each point in the point cloud.

2.1 Voxel grid
Voxel can be understood as a generalization of two-dimensional pixels in three-dimensional space \([8]\), which are a set of cube elements distributed in the centre of orthogonal grids. The process of transforming a geometric (vector) model into a voxel model is called voxelization \([9]\). Generating three-dimensional voxels from laser scan data is a common method for target extraction. This type of method reduces the redundancy of point cloud data, speeds up the calculation, and solves the problem of disordered and unstructured point cloud data applied to deep neural networks. However, most of the current voxel generation methods only consider the spatial neighbour relationship, and do not consider the homogeneity of the points within the voxel, and it is easy to produce mixed voxels. In this paper, when applying 3D-CNN to classify point cloud, the algorithm improves the processing method of point cloud. Unlike the full meshing of the point cloud, the algorithm selects the neighbourhood for each point in the point cloud and voxels the neighbourhood of the point into a voxel grid of size \( n \times n \times n \). The voxel grid for each point serves as the input to the 3D-CNN.

Due to the scene complexity and target diversity of the LiDAR point cloud, single-scale voxels cannot accurately represent the local geometric features of the target. Therefore, when constructing a voxel grid for points in point cloud, this paper selects different neighbourhood sizes as different scales. Multi-scales input 3D-CNN to extract features, and the features of different scales are combined by the fully connected layer to enhance the expression of the local features of the points.

For each mesh in a voxel grid, there are different ways to represent its properties. The easiest way is to calculate the occupancy value of the grid. When there have points in the grid, its value is 1, and when there is no point in the grid, its value is 0. A more complicated method is to calculate the density value of each grid, which can be achieved by taking the grid size and the number of points. In the algorithm of this paper, the occupancy value is used to represent the attributes of each grid. For each point in the point cloud, after selecting \( m \) scales and constructing a voxel grid of size \( n \times n \times n \), it can be represented by a multi-dimensional vector of \( m \times n \times n \times n \). The algorithm of the voxelization algorithm is as follows:

2.2 3D-CNN
This paper classifies LiDAR point cloud data based on the three-dimensional convolutional neural network (3D-CNN) structure. 3D-CNN was first proposed in the application of video analysis \([10]\), because video can be regarded as a three-dimensional extension of the two-dimensional image in time. In the LiDAR field, the literature \([11]\) is an early work to study the use of 3D-CNN to perform binary classification tasks for LiDAR data. These algorithms are similar in network structure, but the nature of the data used is quite different. Due to the disordered and unstructured point cloud, point cloud data needs to be processed when classifying LiDAR data using 3D-CNN. In this paper, the processing method of point cloud voxelization is improved, and the classification of LiDAR point cloud is completed by using 3D-CNN. The network structure used is shown in Figure 2. In the figure, input represents input at different scales, \( C \) represents convolutional layer, \( M \) represents Max pooling layer, \( FC \) represents fully connected layer, \( d \) represents dropout layer, and \( s \) represents softmax layer.
The input to the network is a voxel grid of size $n \times n \times n$. Each point in the point cloud selects five neighbourhoods of different scales and constructs a voxel grid in the neighbourhood. The algorithm in this paper chooses $n=16$, and the value in the grid is determined by the occupancy value. In the experiment, each grid unit can contain additional values, such as intensity information of LiDAR point cloud data, and RGB information from images.

The convolutional layer uses $d$ convolution kernels of size $f \times f \times f$. The size of the convolution kernel in this algorithm is $f=3$. The output of the input voxel grid $(x, y, z)$ at the $t$-th convolution kernel of the $l$-th convolutional layer is:

$$v_{lt}^{(x,y,z)} = b_{lt} + \sum_{q=0}^{q_{l-1}} \sum_{j=0}^{f-1} \sum_{i=0}^{f-1} \sum_{k=0}^{f-1} w_{ijq}^{(l-1,q)} v_{(l-1)q}^{(x+i,y+j,z+k)}$$

Where $b_{lt}$ represents the deviation of the convolution kernel, and $q$ represents the number of convolution kernels of the $l$-1st convolutional layer, $w_{ijq}^{(l-1,q)}$ represents the weight at the $(i,j,k)$ position of the $q$-th convolution kernel, and the deviation and weight are obtained by network training.

The Max pooling layer in the network can be represented by $m(n, g)$. $n$ indicates that the input size of the Max pooling layer is $n \times n \times n$, and $g$ indicates that the size of the pooled kernel is $g \times g \times g$. The output of the input $(x, y, z)$ position at the $t$-th pooled kernel of the $l$-th Max pooling layer is:

$$v_{lt}^{(x,y,z)} = \max_{i,j,k \in [0,1,\ldots,n-g+1]} v_{(l-1)t}^{(x+i,y+j,z+k)}$$

The network consists of two fully connected layers, the first fully connected layer combining features extracted at different scales. The second fully connected layer outputs a 9-dimensional vector, indicating that the points in the point cloud correspond to the scores of the nine categories. Adding a dropout layer between the two fully connected layers prevents overfitting, and the probability of the dropout layer is 0.5.

The Adam optimization algorithm is used for network training. Adam was originally proposed by Diederik Kingma of OpenAI and Jimmy Ba [12] of the University of Toronto. Due to the sparsity of airborne LiDAR point cloud data, when the network updates parameters, the same learning rate cannot be adapted to all parameter updates. Compared with the gradient descent algorithm and its variant Stochastic gradient descent (SGD), the Adam optimization algorithm can improve the efficiency and accuracy of network training when training point cloud data.

At the same time, the neural network uses ReLU as the activation function, and ReLU was proposed in 2016 [13]. The end of the network uses the softmax layer to convert the output of the second fully connected layer to the probability of corresponding category. The cross-entropy loss function is used in training, and the cross entropy loss function is derived from the concept of entropy in information theory.

3. Experimental simulation

This paper uses the German Vaihingen city test data set provided by ISPRS to test the proposed deep neural network. The data set was provided by ISPRS and scanned using the Leica ALS50 airborne LiDAR system. A description of the data set is provided in [14]. The data set contains a rich geographical environment, urban environment and building type, as shown in Figure 3, which can...
fully verify the application of the algorithm in large-scale outdoor scenarios. The LiDAR point cloud data density is 4 points/m³.

Figure 2. Vaihingen city dataset point cloud (coloring by elevation)

In the Vaihingen city dataset, using the algorithm and other neural network algorithms for testing, the semantic tags of the dataset contain 9 categories (Powerlines, Car, Low vegetation, Impervious surfaces, Roof, Fence/Hedge, Facade, Shrub, Tree). The data set contains a total of 753,876 points. This paper selects 301,550 points as the training data set and 452,326 points as the test data set. When constructing voxel grids of different scales, according to the point cloud density of the data set, the five different size neighborhoods selected by the experiment are [3.2m, 4.8m, 6.4m, 8m, 9.6m]. A-sized voxel grid is constructed based on the size of the neighborhood. The classification results are shown in Figure 4. From left to right, the input point cloud, the classification result of the algorithm and the correct classification result.

Figure 3. Classification result of Vaihingen city dataset

4. Experimental results and analysis
In this paper, the classification results are evaluated by using confusion matrix, average classification accuracy and overall classification accuracy. Table 1 shows the confusion matrix of the classification results of the algorithm in this paper. The $i$-th row and the $j$-th column in the confusion matrix represents the percentage of points in the $i$-th row category that are classified into the $j$-th column category, The abscissa of the matrix is the category true value.
Table 1. The confusion matrix of the algorithm classification results in this paper.

| Vaihingen | Powerline | Car | Low vegetation | Impervious surfaces | Roof | Fence/Hedge | Facade | Shrub | Tree |
|-----------|-----------|-----|----------------|--------------------|------|-------------|--------|-------|------|
| Powerline | 23.1      | 3.0 | 1.5            | 15.6               | 21.4 | 5.9         | 19.6   | 5.7   | 4.2  |
| Car       | 0.3       | 45.7| 15.3           | 2.5                | 6.8  | 1.9         | 1.0    | 24.5  | 2.0  |
| Low vegetation | 1.3 | 16.7| 48.6           | 0.5                | 1.4  | 4.0         | 2.2    | 20.7  | 4.6  |
| Impervious surfaces | 0.2 | 0.8 | 8.4           | 78.6               | 3.6  | 0.5         | 2.4    | 4.9   | 0.6  |
| Roof      | 0.1       | 0.5 | 1.6            | 2.0                | 84.4 | 0.6         | 2.3    | 1.7   | 6.8  |
| Fence/Hedge | 3.8 | 4.5 | 12.3          | 0.9                | 3.1  | 33.4        | 27.6   | 8.2   | 6.2  |
| Facade    | 5.1       | 2.7 | 7.3            | 1.2                | 0.8  | 26.5        | 32.5   | 10.4  | 13.5 |
| Shrub     | 1.4       | 18.9| 19.7           | 2.1                | 1.1  | 5.0         | 2.5    | 37.7  | 11.6 |
| Tree      | 0.3       | 1.0 | 2.6            | 1.2                | 6.1  | 0.2         | 4.7    | 2.4   | 81.5 |

As shown in Table 1, impervious surface, roof and tree can achieve higher classification accuracy. However, due to the angle and height of the airborne LiDAR scanning, the points of powerlines, fence/hedge and facade are sparse, the classification is difficult, and the classification accuracy is relatively low. At the same time, low vegetation, shrub and car are similar in height, and the point cloud data used does not contain spectral information such as RGB, which makes these three categories prone to misclassification. There are also some misinterpretations in places where tall trees and buildings meet.

Table 2. Classification accuracy of Vaihingen city dataset.

| Vaihingen dataset | Overall classification accuracy (%) | Average classification accuracy (%) |
|-------------------|-------------------------------------|------------------------------------|
| 3D-CNN[15]        | 69.8                                | 40.6                               |
| PointNet[16]      | 74.2                                | 43.1                               |
| Ours              | 82.6                                | 51.7                               |

Table 2 compares the classification accuracy of the proposed algorithm with the other two neural network algorithms under the Vaihingen urban dataset. It can be seen that the proposed algorithm can achieve higher classification accuracy than the other two neural network algorithms for point clouds. In [15], the whole point cloud is voxel meshed, and mixed voxels appear at a single resolution, which reduces the classification accuracy of the point cloud. The literature [16] uses a symmetric function to make the point cloud directly input into the network without voxelization, but the network is slightly weaker in extracting local features. In this paper, the processing of point cloud is improved, which makes the classification of airborne LiDAR point cloud achieve higher precision. At the same time, because the data used in the experiment does not contain spectral information, the fusion of point cloud data and remote sensing images can further improve the classification accuracy of point clouds [17].

5. conclusion
In this paper, a point cloud classification algorithm based on three-dimensional convolutional neural network is proposed to deal with the classification of ground objects in large-scale complex scenes of airborne LiDAR. Based on the processing of point cloud voxelization, the algorithm improves the processing method of point cloud. Different from the voxelization of all point clouds into voxel grids, the algorithm constructs voxel grids of different scales for each point in the point cloud. The voxel grid where each point is located is used as an input, and algorithm apply 3D-CNN for feature
extraction to complete the classification of the LiDAR point cloud. By testing on the Vaihingen city dataset, the proposed algorithm can achieve higher classification accuracy than other neural network algorithms. However, due to the height and angle of the airborne LiDAR scanning, the airborne LiDAR point cloud is more sparse, and there are many mis-points in the experimental results, especially for the power line and other categories, the classification is more difficult. We need further improvement or fusion with remote sensing images to improve classification accuracy.

Acknowledgments
This work is supported by National Science Foundation of China under Grant 41601436

References
[1] Koppula H S, Anand A, Joachims T, et al. Semantic labeling of 3D point clouds for indoor scenes[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2011:244-252.
[2] Song S, Xiao J. Sliding Shapes for 3D Object Detection in Depth Images[C]// European Conference on Computer Vision. Springer, Cham, 2014:634-651.
[3] Kanai S. Detection and Classification of Pole-like Objects from Mobile Laser Scanning Data of Urban Environments[J]. International Journal of Cad/cam, 2013, 13(2).
[4] Wu Z, Song S, Khosla A, et al. 3D ShapeNets: A deep representation for volumetric shapes[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014:1912-1920.
[5] Maturana D, Scherer S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015:922-928.
[6] 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.
[7] 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.
[8] Kaufman A, Cohen D, Yagel R. Volume Graphics[J]. Computer, 1993, 26(7):51-64.
[9] Guan W, Lin X, Ma S. Deformable registration of digital images[J]. Journal of Computer Science & Technology, 1998, 13(3):246-260.
[10] Karpathy A, Toderici G, Shetty S, et al. Large-Scale Video Classification with Convolutional Neural Networks[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1725-1732.
[11] Prokhorov D. A convolutional learning system for object classification in 3-D Lidar data[J]. IEEE Transactions on Neural Networks, 2010, 21(5):858-863.
[12] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
[13] Shang W, Sohn K, Almeida D, et al. Understanding and improving convolutional neural networks via concatenated rectified linear units[C]//International Conference on Machine Learning. 2016: 2217-2225.
[14] Rottensteiner, Franz, et al. "ISPRS test project on urban classification and 3D building reconstruction." Commission III-Photogrammetric Computer Vision and Image Analysis, Working Group III/4-3D Scene Analysis (2013): 1-17.
[15] Huang J, You S. Point cloud labeling using 3D Convolutional Neural Network[C]// International Conference on Pattern Recognition. IEEE, 2017.
[16] Charles R Q, Su H, Mo K, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[J]. 2016:77-85.
[17] Dong B G. Research on Feature Classification Technology of Airborne LiDAR Point Cloud and Remote Sensing Image Fusion[D]. Information Engineering University,2013.