ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

Research on Building Facade Feature Information Extraction

Junli Yang, Yongqiang Li, Xiang Pan, Jing Zang

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

Abstract: *As one of the main components of the urban interior, buildings play a pivotal role in the construction of smart cities. In order to segment the structural characteristics contained in building facades more accurately and efficiently and intelligently, this paper constructs a small building facade dataset based on on-board LiDAR point clouds, and improves it based on the original PointNet++ network by adding a multi-headed attention mechanism module in model feature extraction; in model training, an extraction function is introduced after the training function for unlabelled In terms of model training, an extraction function is introduced after the training function to extract features from the unlabelled building façade data, and the extraction results are finally given labels and saved to the training data set, so as to achieve the effect of enhancing the training sample data, and by improving the model structure and model training method, the effective extraction of building façade features is achieved. The final accuracy of the proposed method is 88.2% and the average intersection-to-merge ratio (mIoU) is 77.1%, which is 1% and 3.3% higher than the original PointNet++ network respectively.*

Keywords: *deep learning; PointNet++; In-vehicle LiDAR point cloud; building façade; feature extraction*

1. Introduction

With the successive concepts of smart earth and smart city, there are higher requirements for the construction of fine 3D models of buildings, and whether the internal feature information of the façade, as an important part of the building, can be extracted finely directly affects the final construction accuracy of the building model. In recent years, the traditional algorithm in the extraction of building façade point cloud feature information has been quite in-depth research[1-5], the extraction of higher and higher accuracy, however, its shortcomings are: (1) need to artificially find the laws and features contained within the object of study; (2) extraction process is relatively complex, need to interactively set multiple parameter values, the degree of intelligence is relatively low, the universal applicability is relatively poor. With the gradual maturation of deep learning in processing 2D data, research workers gradually introduced point cloud data into the framework of deep learning for processing, initially indirectly converting 3D irregular point cloud data into regular 2D data, such as projection-based methods[6-9], voxel-based methods[10-14], and multi-view-based methods[15-17], which have the advantage that they can apply mature deep learning network models for processing, with the disadvantage that some important feature information is lost during the data transformation process. Because of the irreplaceable advantages of 3D point cloud data over image data, deep learning based directly on point cloud data has gradually become a hot topic of research for scholars at home and abroad[18,19]. At present, deep learning is mainly based on public datasets, and its practical application is more for classification and segmentation of buildings with large scenes of airborne LiDAR point clouds[20-23], while relatively little research has been conducted on the extraction of building façade point cloud features.

Most of the current research scholars' improvement strategies for deep learning networks are aimed at the network structure of the model, through methods such as structural optimization or substitution, so as to improve the accuracy of the model's results in processing data[24-27]. In this paper, based on the original PointNet++[19] network model, we first add a multi-headed attention mechanism to the feature extraction part of the model to further improve the network's ability to extract feature information, and improve the training part of the model on this basis by introducing unlabelled elevation datasets, which are automatically added to the training dataset after multiple cycles of network processing, to complete the training dataset. This can largely reduce the workload of constructing the dataset and effectively improve the accuracy of the network in extracting the feature information of building facades.

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

2. Principle of the algorithm

2.1 PointNet++ network model improvement strategies

In order to further improve the accuracy of the network in the extraction of building façade feature information, this paper proposes an improved version of the PointNet++ network, and the overall structure of the network model is shown in Figure 1. The network is internally divided into two parts: encoding and decoding. The overall computational process of the network is shown in Figure 2, where the input data is used for local feature extraction through the SA1 module, in which the number of original point cloud data is first downsampled from 2048 to 512 using Farthest Point Sampling (FPS), and the 512 points obtained are used as the centroids of the local area. The SA2 module takes the results obtained from SA1 as input, downsamples the number of points from 512 to 128 using FPS, groups them with a radius of 0.4 and a maximum of 64 nearest neighbours. The obtained feature information is passed to the multi-headed attention mechanism module and the maximum pooling module respectively, and the results of both are fused as the feature information of each centroid; the output of SA2 module is used as the input of SA3, which divides the input 128 points into a group and extracts the global feature information through PointNet feature information.

In the process of decoding, the network first passes the results obtained from SA2 and SA3 into FP1, finds the three points with the shortest distance to each of the 128 points inside SA2 inside SA3, and after linear interpolation processing of the three point features queried, the 128 points obtain the corresponding interpolated feature information, splices the feature information of the original SA2 with it, and after MLP processing, the point features of FP1 are obtained; afterwards, the results obtained from SA1 and FP1 layers are passed into FP2 layer, and after the same process, the output results of FP2 are obtained; the original point set and FP2 results are passed into FP3 as input, and the feature information is obtained after linear interpolation, and the feature information of the original point set is spliced with it, and the point features of each point are obtained after MLP processing; finally, the fully connected layer is used to process Finally, the features obtained by using the fully connected layer are processed to achieve the predictive segmentation of each point.

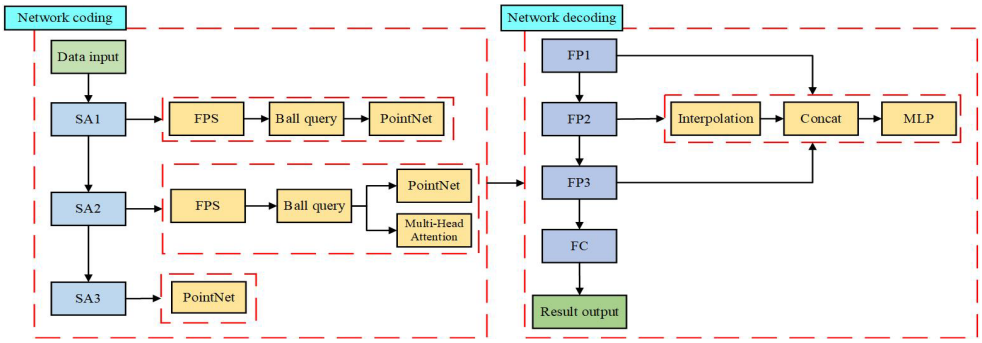


Figure 1: Network model structure flow chart

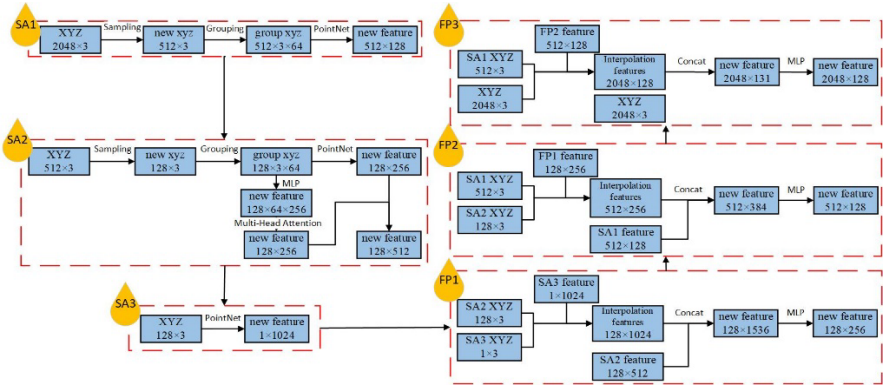


Figure 2: Overall network calculation flow chart

2.2 Principle of the multi-headed attention mechanism

The attention mechanism can enable deep learning networks to focus on information that is more

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

critical to the current task from a large amount of input and reduce the attention to other information, or even filter out irrelevant information, thus improving the efficiency and accuracy of task processing, however, in the actual processing, the attention mechanism only assigns a single weight to each feature vector, and the model cannot emphasize certain features of the points, so this paper introduces the multi-head attention mechanism[28].

As an upgraded version of the attention mechanism, the core computational module of the multi-headed attention mechanism is still the attention mechanism, but the difference is that the parameters of each attention operation of the multi-headed attention mechanism are independent of each other. Firstly, Q, K and V are mapped into a higher dimensional space through a fully connected layer, and Q, K and V of the same dimension are divided into groups, and each group is passed into the attention mechanism separately for computation, and the output of multiple groups is finally stitched together as the final result output, where the single-headed computation is shown in equation (1).

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{1}$$

Where: W_i^Q represents the weight matrix corresponding to the query matrix; W_i^K represents the weight matrix corresponding to the key matrix; W_i^V represents the weight matrix corresponding to the value matrix.

When the data is passed into the attention mechanism, the transpose of the query matrix Q and the key matrix K are first dotted and divided by the scaling factor to obtain the attention scores of all points after which the attention vectors are transformed into weights by a softmax operation and multiplied with the value matrix V. as shown in equation (2); the multiple heads are finally stitched together as the final result, as shown in equation (3).

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}\left(\text{head}_1,\ldots,\text{head}_h\right) \tag{3}$$

2.3 Building a multi-headed attention mechanism

In addition to XYZ coordinate information, the vehicle LiDAR point cloud data also contains a large amount of feature information, such as normal vectors, colour and reflection intensity, so the initial feature vectors of the point cloud data are expressed as shown in Equation 4:

$$q_i = \{x_i, y_i, z_i, \ldots, p_i\}, i = 1,2,\ldots,N \tag{4}$$

where q_i denotes a single point within the point cloud data.

After determining the query matrix Q, the key matrix K and the value matrix V, they are mapped into the high-dimensional space through multiple fully connected layers, respectively, after which they are passed into the attention mechanism, respectively, and the attention score of the query vector is calculated using the scaled dot product model, as shown in Eqs. 5:

$$s(q_i q_j) = \frac{q_i q_j^T}{K} \tag{5}$$

where q_i denotes the query point, q_j denotes any point inside the point set, and K denotes the scaling factor.

The attention scores of the query points for each point are obtained, the attention vectors are transformed into weights using the softmax function, and finally the final result of q_i is calculated by using Eqs. 6.

$$q_i = \sum_{j=1}^{N} w_{ij} q_j \tag{6}$$

where w_{ij} is used as the weight of the query point q_i on the input point q_j.

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

In this paper, the local region feature information output from a multilayer MLP is processed by constructing a four-headed attention mechanism with a single local region as an example and the input matrix shown in Eqs. 7:

$$F_{N\times 256}=\begin{bmatrix} x_1 & y_1 & \cdots & p_1 \\ x_2 & y_2 & \cdots & p_2 \\ \vdots & \vdots & \cdots & \vdots \\ x_N & z_N & \cdots & p_N \end{bmatrix} \tag{7}$$

where Eq. N is the number of point clouds and 256 is the feature dimension.

Q uses the centroid of the matrix containing the point cloud as the query vector, and K and V use the F matrix as the key vector and value vector respectively, after which the four fully connected layers are used to map into 512, 768, 1024 and 1280 high-dimensional spaces respectively, and the results are obtained and passed into the attention mechanism respectively, with each set of computational equations shown in 8.

$$\begin{aligned} h_{head1}&=softmax\left(\frac{QK^T}{\sqrt{N_1}}\right)\cdot F_{N\times 512} \\ h_{head2}&=softmax\left(\frac{QK^T}{\sqrt{N_1}}\right)\cdot F_{N\times 768} \\ h_{head3}&=softmax\left(\frac{QK^T}{\sqrt{N_1}}\right)\cdot F_{N\times 1024} \\ h_{head4}&=softmax\left(\frac{QK^T}{\sqrt{N_1}}\right)\cdot F_{N\times 1280} \end{aligned} \tag{8}$$

where N1 is the dimensional value of the matrix Q.

Once the results for each head have been calculated, they are stitched together as shown in equations 9:

$$F_{attention}(H)=concat(h_{head1},\cdot h_{head2},\cdot h_{head3},\cdot h_{head4}) \tag{9}$$

Afterwards, the $F_{attention}(H)$ is mapped using the fully connected layer as the final output, and the complete structure flow is shown in Figure 3.

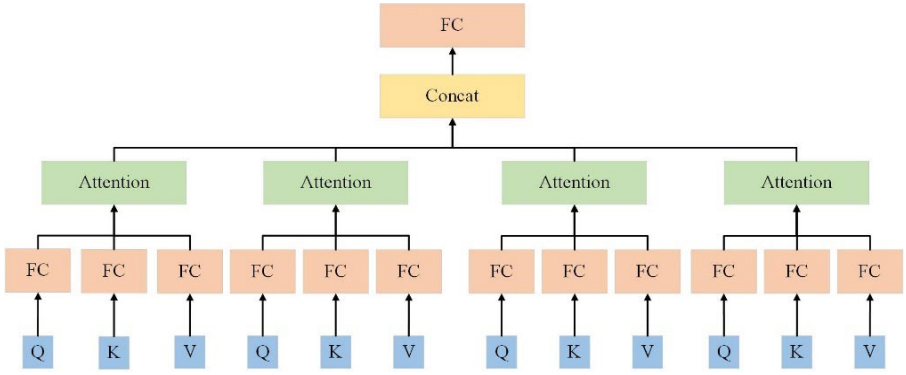


Figure 3: Flow chart of the structure of the four-headed attention mechanism

2.4 Network model training improvement strategies

The main reason is that the training process of the network model requires a large amount of sample data to enable the network to learn various types of feature information within the samples, which makes it costly to build the dataset. This paper constructs a small building façade dataset based on the research object. By improving the original training mode of the network model and introducing unlabelled façade data, the network automatically makes the unlabelled data into the required sample data during the training process, thus compensating for the impact of the number of training samples on the network.

The basic idea of the improved network model training model proposed in this paper is to introduce a feature extraction function after the original training function, which is based on the original test code and modified so that it can directly extract the internal feature information from the introduced unlabelled

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

data after the training of the network model, and the extracted results are processed twice to generate samples that meet the requirements of the network, and are automatically added to the training data set as the sample data for the next cycle of training. The result will be automatically added to the training data set as the sample data for the next training cycle, thus achieving a rapid expansion of the samples, thus solving the problem of a small number of samples in the data set, and ultimately improving the accuracy of the network in extracting the feature information of the building facade. The specific implementation is as follows:

(1) Firstly, the training data in the dataset constructed in this paper is passed into the deep learning network for the training of the model. The network model learns the attributes and features of each category within the building façade based on the label information contained in the data through continuous iterative training, so that the network model can initially grasp the ability to distinguish various types of feature information within the building façade.

(2) Extract the X, Y and Z coordinate values from the saved results, and use the algorithm to calculate the normal vector information of the coordinates; for the automatic label assignment problem, it is mainly based on the colour information contained in each point in the output results, and assign the corresponding category label information through the colour information, and finally integrate the coordinates, normal vector and label information as sample data to be saved into the training data set, and regenerate the training data set into the network in json format.

(3) Repeating steps (1), (2) can quickly expand the sample data needed for training, but because the data added to the training set in step (2) is the same region each time, it is easy to make the network overfitting problem, in order to prevent the model from this problem, in this paper, before adding the produced sample data to the training dataset automatically, the point cloud in the sample data is randomly panned, rotated and In order to prevent this problem, this paper performs random translation, rotation and scaling processes on the point clouds in the sample data before adding them to the training dataset automatically to change their original spatial positions and finally achieve data enhancement operations.

3. Experimental analysis

3.1 Experimental environment

In this paper, a DELL G3 3590 laptop was chosen as the experimental platform, with an internal CPU of i5-9300H, a graphics card of Nvidia Geforce GTX 1650 and 4GB of graphics memory; the environment configuration and operation of the deep learning network was carried out under Ubantu 16.04 system, and a virtual environment was created through Anaconda, in which In the virtual environment, the appropriate libraries necessary for the network were installed, such as Python 3.7, Tensorflow-GPU 1.13, H5py and Opencv-Python 3.4.4.19, and CUDA 10.1 and Cudnn 7.5 were installed to accelerate the GPU computation.

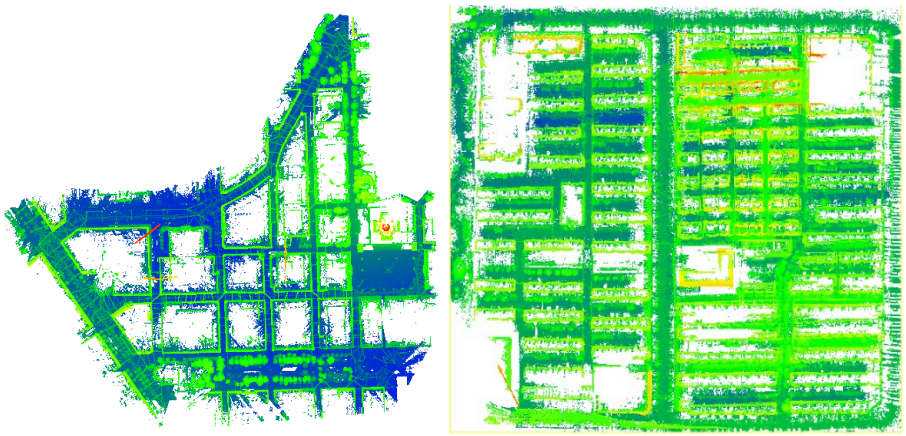
3.2 Experimental data

This experiment selects two sets of in-vehicle LiDAR point cloud data sets, data set 1 is a local area of a foreign city point cloud data production samples, the region has a total of 28 buildings, the area is about 0.03km^2, the vegetation in the region is relatively small, the overall height of the buildings are lower than 20 meters, the elevation point cloud data is relatively complete, in-vehicle point cloud data as shown in Figure 4 (a); data set 2 is a residential district in Zhengzhou There are 105 buildings in the district, with an area of about 0.32km^2, obtained by the SSW vehicle-mounted LiDAR system, as shown in Figure 4(b); all buildings in the district are below 20m, except for four buildings with floors higher than 100m, and the tops of the buildings are mostly flat or pitched, and the elevations are flat and the angle between adjacent elevations is right.

The original in-vehicle LiDAR point cloud data is very complicated, which contains various kinds of feature point clouds and a large number of noisy points. In this paper, we take the building façade as the research object, and need to extract the building façade object in the large scene point cloud, so we choose the method proposed in the literature [29] to extract the building façade point cloud from the large scene original point cloud data. After obtaining the building façade point cloud data, the radius filtering algorithm was used to remove the noise points on it, and the Open3D library in python was used to obtain the normal vector information of the point cloud; then some of the façades were selected to segment the internal structure of the façade by human-computer interaction and given the corresponding label

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

information, and the sample data of the building façade was obtained after normalisation and downsampling. The sample data were then randomly selected and grouped by python program in the ratio of 6:2:2 to form the training set (Figure 5), validation set (Figure 6) and test set (Figure 7); finally, the remaining unlabelled façade data were collated to generate the data set to be extracted.



(a) In-vehicle data for a foreign city *(b) Vehicle-mounted data for a district in Zhengzhou*

Figure 4: Vehicle LiDAR raw point cloud data

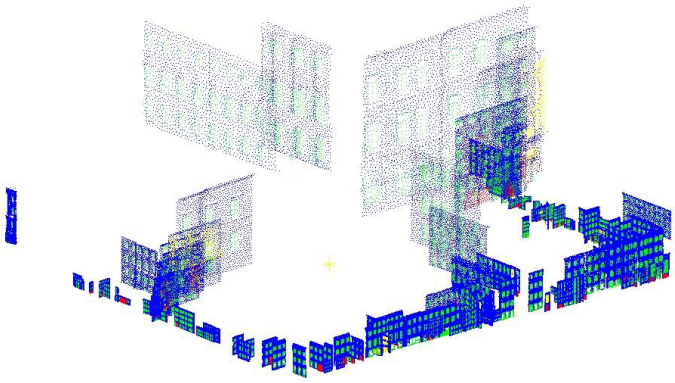


Figure 5: Training dataset display

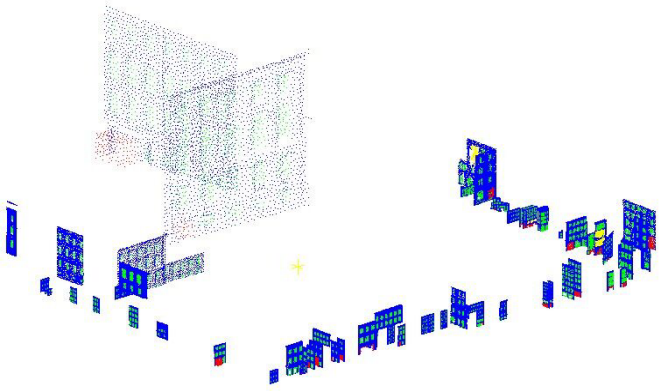


Figure 6: Validation dataset display

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

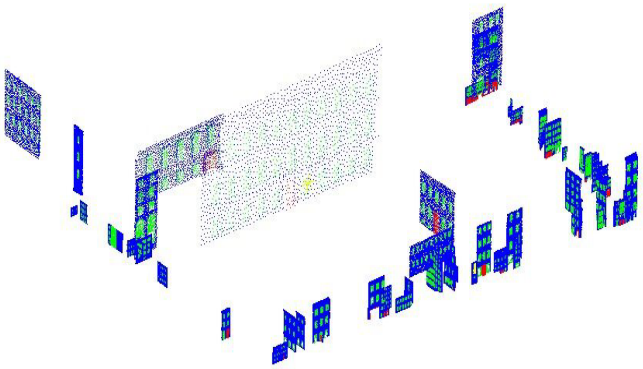


Figure 7: Test data set display

3.3 Model training

The improved PointNet++ model increases the amount of data in the training set as the number of cycles increases during the training process. The corresponding hyperparameters in the network are set as follows: batch_size is 6, initial decay rate is 0.7, decay rate is 200000, initial learning rate is 0.001, optimisation method is adam, momentum is 0.9, num_point is 2048, max_epoch is 201, and the number of model training cycles is 3. The network model in In the last training and learning process, the overall trend of the loss value was decreasing as the number of iterations increased, and the overall loss value was smaller than the original network training loss value compared with the original network, and the final fluctuation was stable between 0.011-0.012, while the original network fluctuation was stable between 0.013-0.014, as shown in Figure 8; the overall trend of the improved network in the training Compared with the original network, most of its accuracy values were greater than the accuracy values of the original network, and the overall fluctuation was smaller than that of the original network, thus proving that the improved network was more stable than the original network, as shown in Figure 9.

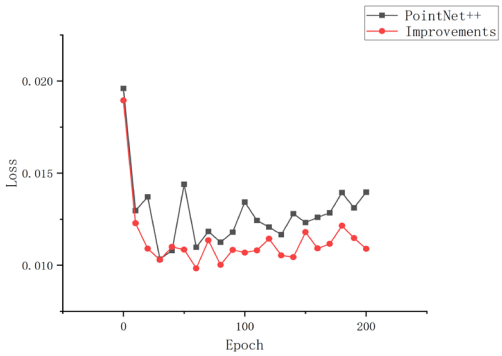


Figure 8: The evolution of loss values during training

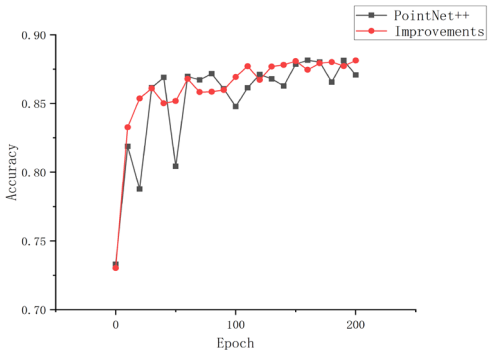


Figure 9: The evolution of accuracy during training

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

3.4 Comparison and analysis of network extraction results

In order to verify the effectiveness of the method proposed in this paper, several deep learning networks were selected to train and test the model on the building façade dataset constructed in this paper, and their extraction accuracy was compared with the accuracy of the results obtained by the method proposed in this paper. The final extraction results of each network are shown in Table 1, and the extraction results of each network for each type of building façade are shown in Table 2.

Table 1: Accuracy of extraction results across networks

Deep Learning Networks	Accuracy (%)	mIoU (%)
PointNet	73.2	58.9
PointNet++	87.2	73.8
Improvements	88.2	77.1

Table 2: Mis-extraction statistics by category for each network dyad

Networks	Wall (errors/total)	Windows (errors/total)	Balcony (errors/total)	Door (errors/total)
PointNet	0/119849	37691/37691	2198/2198	4102/4102
PointNet++	9464/119788	6048/37765	2191/2216	3345/4071
Improvements	6922/119915	8141/37705	1549/2213	2755/4007

In Table 1, we can see that the extraction accuracy of the proposed method is 88.2% and the average cross-merge ratio is 77.1%, which is 15% and 18.2% higher than the PointNet network, and 1% and 3.3% higher than the original PointNet++ network, and the overall accuracy of the improved network is improved compared with the original network; In Table 2, it is found that the PointNet network classifies each category of feature information as wall, and its misclassification rate is 0, 1, 1 and 1 for wall, window, balcony and door respectively, and the misclassification rate of PointNet network is 7.9%, 16%, 98.9% and 82.2% for each category of feature information respectively, and the misclassification rate of the improved network is 5.8%, 21.6%, 70% and 68.8% respectively, 21.6%, 70% and 68.8% respectively. By comparing the error rate of the original network with the improved network, the error rate of the improved network was reduced by 2.1% compared to the pre-improved network in terms of walls, increased by 5.6% in terms of windows, reduced by 28.9% in terms of balconies and reduced by 13.4 in terms of doors, the accuracy of the improved network in extracting balconies and doors was The accuracy of the improved nets in extracting balconies and doors has been greatly improved, but the accuracy in window extraction has been reduced , as shown in Figure 10, Figure11 and Figure 12.

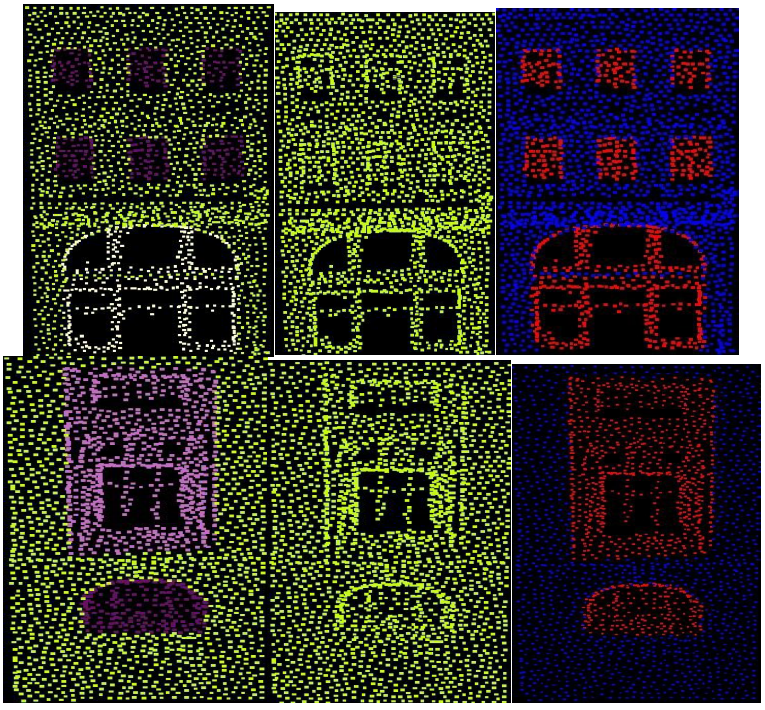


Figure 10: Partial display of PointNet network extraction results

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

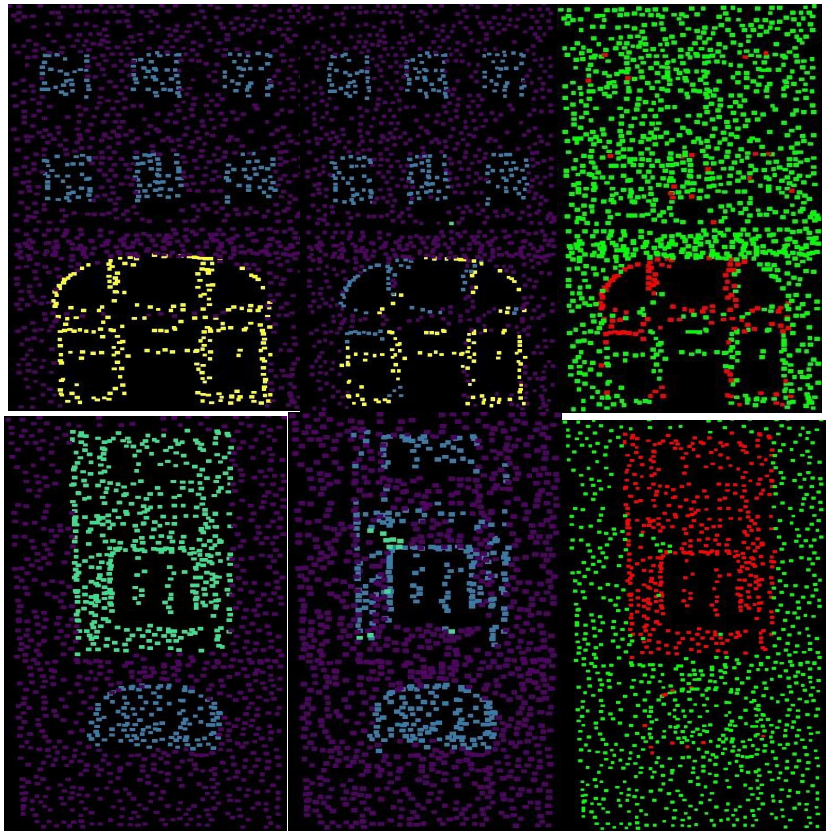


Figure 11: Partial display of PointNet++ network extraction results

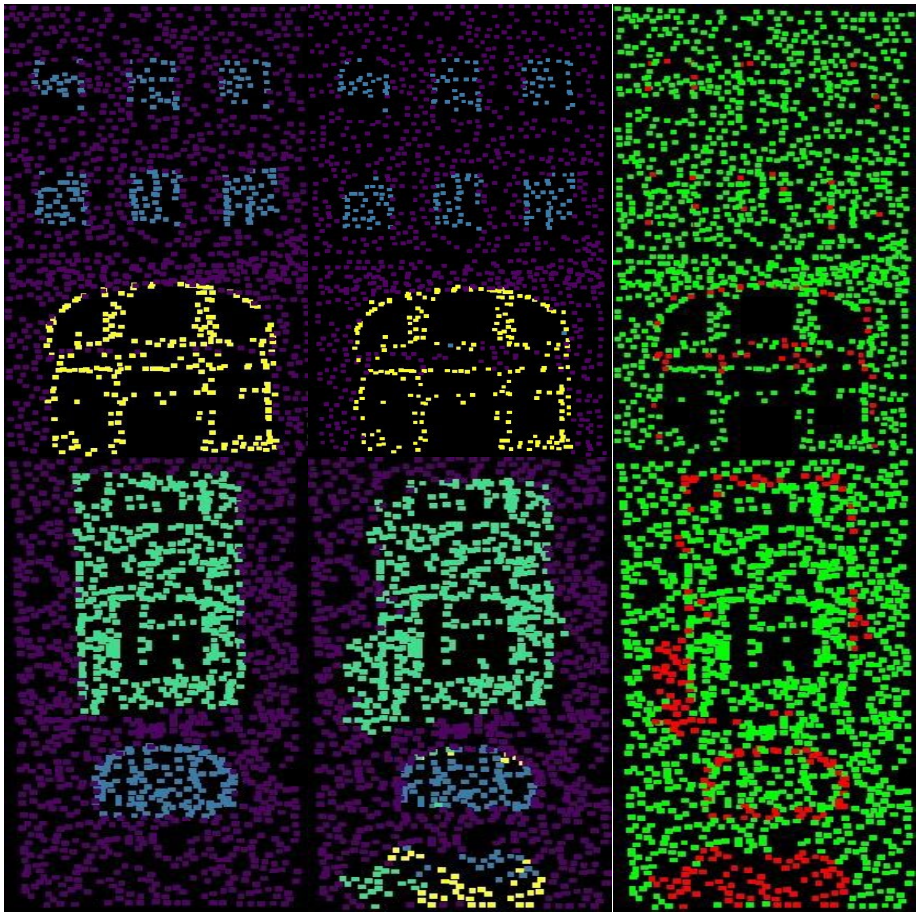


Figure 12: A partial demonstration of the extraction results of the method in this paper

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

3.5 Analysis of the number of cycles

In this paper, based on model improvement, we propose an enhanced operation of training data based on unlabeled data. The method uses a small amount of building façade data containing label information to train the network model, so that the network can learn the feature information of each category in the façade, and then introduces the unlabeled façade data set, and uses the model obtained after training to extract feature information from the unlabeled data set. This method allows the unlabelled data to be roughly labelled and the errors to be ignored, and the labelled data to be automatically added to the training data set to increase the amount of training data for subsequent networks, thus achieving data enhancement. In this section, the number of training cycles is specified as 1, 2, 3, 4 and 5, and the final accuracy of the results is compared to determine the optimal number of cycles.

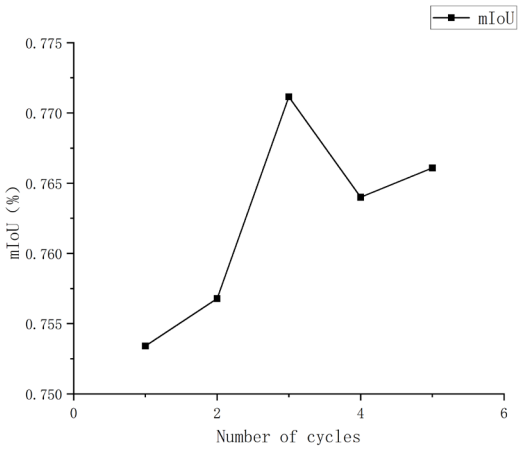


Figure 13: Results statistics for different cycle times

The results were obtained after several trials, as shown in Figure 13, when the number of training cycles in the network was 1, the network training process was not enhanced, and the network for building facade feature information extraction mIoU was 75.3%; when the number of cycles was 2, the training data was enhanced, and the mIoU result was 75.7, compared with the first cycle mIoU increased by 0.4%; when the cycle 3 times, the mIoU result was 77.1%, an increase of 1.8% and 1.4% compared to the first and second cycle results, respectively; when cycled 4 times, the mIoU result was 76.4%, an increase of 1.1% and 0.7% compared to the first two cycles, respectively, and 0.7% lower than the third cycle; when cycled 5 times, the mIoU result was 76.6%, higher than the first 1.3%, 0.9% and 0.2% higher than the 1st, 2nd and 4th cycles respectively, and 0.5% lower than the 3rd cycle; after comparing and analysing the multiple extraction results, it is thus concluded that when the number of cycles is set to 3, its boost to the network is relatively the highest.

3.6 Ablation experiments

This paper improves on the original PointNet++ network by introducing a multi-headed attention mechanism and changing the model training mode. The final extraction results are obtained after experimental analysis, and the extraction accuracy of the improved network is improved compared to the original network, while ensuring that the basic parameters are set consistently within the network. In this subsection, in order to investigate the influence of each module in the network on the extraction results, it is proposed to test the degree of influence of different combinations of each module of the network on the final extraction accuracy by reducing the number of modules.

Table 3: Ablation experiments

Training mode	Multi-headed attention	Accuracy (%)	mIoU (%)
√	√	88.2	77.1
×	√	88.5	75.3
√	×	87.9	74.2
×	×	87.2	73.8

From Table 3, it can be found that when the multi-headed attention mechanism is added to the original PointNet++ network model and the model training mode is changed, the final extraction accuracy of the network is 88.2% and the final result of mIoU calculation is 77.1%; when the model training mode is not

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

changed in the network, i.e. the multi-headed attention mechanism is added to the network alone, the final extraction accuracy of the network is 88.5% and the mIoU result is 75.3, which is a 0.3% increase in accuracy and a 1.8% decrease in mIoU compared to the overall network with the addition of the multi-headed attention mechanism and the change in training mode; when the multi-headed attention mechanism module is removed, i.e. when only the model training mode is changed in the network, the final extraction accuracy of the network is 87.9% and the mIoU result is 74.2%, compared to the overall network with the addition of the multi-headed attention mechanism and the change in training mode. Compared to the overall added network, the accuracy decreased by 0.3% and the mIoU result decreased by 2.9%; in the case where neither was referenced, i.e. the final extraction accuracy of the original network was 87.2% and the mIoU result was 73.9, compared to the overall added network, the network accuracy decreased by 1.0% and the mIoU result decreased by 3.3%; the ablation experiments were used to The obtained results were compared and analysed by ablation experiments, and it was found that with the successive introduction of the training mode and the multi-headed attention mechanism, the final mIoU results of the network would be improved by different magnitudes, and the improvement to the network was the greatest when the training mode and the multi-headed attention mechanism modules were added at the same time.

4. Conclusion

Based on the original PointNet++ network model, this paper proposes to add the multi-headed attention mechanism to the feature extraction module to improve the network's ability to extract feature information, and to improve the final extraction accuracy of the network by improving the original training method of the model and adding the unlabelled elevation point cloud data to the model segmentation process through continuous iterative cycles. The experimental results show that the final average extraction accuracy of the proposed method is 77.1%, which is a certain improvement compared with the original network, but there is still room for improvement in the final extraction accuracy of the improved network model. Therefore, further research and optimisation of the network model is needed.

References

[1] Yang Qingke, Li Yongqiang, Liu Cong, et al. Study on Window Boundary Extraction Method of Introducing Dynamic Ellipse Based on Vehicle LiDAR[J]. Geography and Geo-Information Science, 2019, 35(02):61-67.

[2] Wang Y, Ma Y, Zhu A, et al. Accurate facade feature extraction method for buildings from three-dimensional point cloud data considering structural information [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 139: 146-153.

[3] Arachchige N H, Perera S N, Maas H G. Automatic processing of mobile laser scanner point clouds for building facade detection [J]. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 2012, 39: 187-192.

[4] Li Hongmei, Zhang Chunchao, Zhang Xia, Luo Zhu. Feature extraction of building point clouds based on Morse theory [J]. Bulletin of Surveying and Mapping, 2020(05):31-35+42.

[5] Fan Huilong, Li Yongqiang, Yang Qingke, et al. Building façade window model construction of vehicle point cloud data [J]. Bulletin of Surveying and Mapping, 2020, No. 518(05):95-100. DOI: 10. 13474/ j. cnki. 11-2246. 2020. 0153.

[6] Wu B, Wan A, Yue X, et al. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 1887-1893.

[7] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0. 5 MB model size[J]. arXiv preprint arXiv:1602. 07360, 2016.

[8] Wu B, Zhou X, Zhao S, et al. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 4376-4382.

[9] Imad M, Doukhi O, Lee D J. Transfer learning based semantic segmentation for 3d object detection from point cloud [J]. Sensors, 2021, 21(12): 3964.

[10] 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.

[11] Verdoja F, Thomas D, Sugimoto A. Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding[C]//2017 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2017: 1285-1290.

ISSN 2616-5775 Vol. 6, Issue 7: 72-83, DOI: 10.25236/AJCIS.2023.060711

[12] Li Y, Pirk S, Su H, et al. Fpnn: Field probing neural networks for 3d data [J]. Advances in neural information processing systems, 2016, 29.
[13] Tchapmi L, Choy C, Armeni I, et al. Segcloud: Semantic segmentation of 3d point clouds[C]//2017 international conference on 3D vision (3DV). IEEE, 2017: 537-547.
[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] 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.
[16] Jiang J, Bao D, Chen Z, et al. MLVCNN: Multi-loop-view convolutional neural network for 3D shape retrieval [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 8513-8520.
[17] Tatarchenko M, Park J, Koltun V, et al. Tangent convolutions for dense prediction in 3d [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 3887-3896.
[18] Qi C R, Su H, Mo K, et al. PointNet: Deep learning on point sets for 3d classification and segmentation[C]. IEEE, 2017, 652-660.
[19] 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.
[20] Lai Ming, Zhao Jiankang, Liu Chuanqi, et al. Semantic Segmentation of LiDAR Point Cloud Based on CAFF-PointNet [J]. Laser & Optoelectronics Progress, 2021, 58(20):488-497.
[21] Hu Chuanwen, Lu Shijie, Yang Wenjing, et al. Deep learning architecture for building extraction using LiDAR point clouds[J]. Bulletin of Surveying and Mapping, 2021, No. 537(12):88-93.
[22] Dai Mofan, Xing Shuai, Xu Qing, et al. Semantic segmentation of airborne LiDAR point cloud based on multi-feature fusion and geometric convolution[J]. Journal of Image and Graphics, 2022, 27(02):574-585.
[23] Meng Congtang, Zhao Yindi, Han Wenquan, et al. RandLA-Net-based detection of urban building change using airborne LiDAR point clouds[J]. Remote Sensing for Natural Resources, 2022, 34(04):113-121.
[24] Wu Jun, Cui Yue, Zhao Xuemei, et al. SSA-PointNet++:A Space Self-Attention CNN for the Semantic Segmentation of 3D Point Cloud[J]. Journal of Computer-Aided Design & Computer, 2022, 34(03):437-448.
[25] Jiang M, Wu Y, Zhao T, et al. Pointsift: A sift-like network module for 3d point cloud semantic segmentation [J]. arXiv preprint arXiv:1807. 00652, 2018.
[26] Liang Zhenhua, Wang Feng. Attention weighted feature aggregation PointNet network for part segmentation [J/OL]. Application Research of Computers: 1-8 [2023-01-27].
[27] Qian G, Hammoud H, Li G, et al. Assanet: An anisotropic separable set abstraction for efficient point cloud representation learning [J]. Advances in Neural Information Processing Systems, 2021, 34: 28119-28130.
[28] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
[29] Yang Qingke, Li Yongqiang, Li Lixue, et al. Building façade modeling combining Vehicle-borne LiDAR data with airborne point cloud data[J]. Science of Surveying and Mapping, 2019, 44(02):94-101.