MBBOS-GCN: minimum bounding box over-segmentation—graph convolution 3D point cloud deep learning model

Delong Zhan,^{a,b} Dongdong Liang,^{a,b,*} Huiyu Jin,^c and Xu Wu^d

^aAnhui Normal University, School of Geography and Tourism, Wuhu, China

^bThe Second Hydrological Geological Prospecting Institute of Anhui Provincial Bureau of Geological Survey, Wuhu, China

^cFudan University, Department of Environmental Science and Engineering, Shanghai, China

^dAnhui Normal University, School of Computer and Information, Wuhu, China

Abstract. Point cloud data with high accuracy and high density is an important data source for the depiction of real ground objects, and there is a broad research prospect of using point cloud data directly for 3D object detection and recognition using deep learning methods. However, many deep learning models in previous research ignored the point cloud structure information and the sampling randomness. To overcome this limitation, we proposed an innovative 3D point cloud deep learning model, namely, the minimum bounding box over-segmentation—graph convolution 3D point cloud deep learning network model (MBBOS-GCN) for enhancing the structural information perception capability of the model and reduce the sampling randomness. In MBBOS-GCN, the number of points sampled is used as the scale, and a modified graph convolution model is used to collect point cloud structure information from different scales. The point cloud is divided into several small regions by the minimum bounding box algorithm, and the farthest point sampling (FPS) algorithm is used to sample within each small region to reduce sampling randomness. The experiments on object classification and semantic scene data segmentation show that: (1) the MBBOS-GCN model has high classification and segmentation accuracy, which is up to 91.87% and 89.5% on the ModelNet40 dataset and ScanNet dataset, respectively; (2) the MBBOS-GCN model is provided has good stability and robustness with a little change in accuracy under the altering density of input point cloud data, and slight classification loss value; (3) the MBBOS-GCN model can be adapted to real complex scenes when the classification accuracy reaches up to 97.53%. These superior performance of the MBBOS-GCN model can provide an effective support for the construction of digital twin city background data and the calibration of multimode satellite feature inversion algorithm validation. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JRS.16.016502](https://doi.org/10.1117/1.JRS.16.016502)]

Keywords: 3D target recognition; PointNet++; graph convolution; minimum bounding box over segmentation.

Paper 210366 received Jun. 10, 2021; accepted for publication Dec. 29, 2022; published online Jan. 17, 2022.

1 Introduction

Computer vision is able to perceive and recognize the world through acquiring information from sensors instead of human beings, which has been a research hotspot for a long time.^{1–4} Target recognition is a fundamental and important research topic in the field of computer vision that is widely used in reverse engineering, intelligent surveillance, and remote sensing.^{5–8} Compared with two-dimensional (2D) target recognition, how to recognize the position, shape, and pose of three-dimensional (3D) objects in space is more meaningful for practical application scenarios, which is particularly in areas such as unmanned systems and augmented reality technology.^{9,10} 3D target recognition can be divided into three types according to the data source: first, is based

*Address all correspondence to Dongdong Liang, LQF121@ahnu.edu.cn

on single-view 3D target recognition. Under a specific type of target, relying on monocular cameras for 3D detection of objects can give approximate information on the position and size of the object, which is greatly time and cost saving.¹¹ However, the lack of depth information in the monocular view limits the accuracy of recognition, especially if the recognized object is obscured or far away.^{12–14} The second one is 3D target recognition using multi-view images.^{15–17} Through reasonable spatial matching, the relative positions between cameras are calculated to obtain more precise spatial relationships than in the single-view case, and the recognition of 3D targets is achieved by combining existing a priori knowledge.¹⁸ Although multi-view 3D target recognition relatively improves recognition accuracy, the real world is 3D, and 3D data can better directly show the scale, shape, spatial location, and other information of 3D objects. The third kind of target recognition methodology with 3D data has become a research hotspot in computer vision research.¹⁹

The 3D point cloud data is a data set consisting of a series of disordered points with high dimension data information,^{20,21} which is an important data source in 3D data and widely used as input data for 3D object recognition.^{22,23} Currently, there are two main approaches to use point cloud data to identify and detect 3D objects: (1) pre-processing the point cloud data first, and then using the pre-processed data to detect and identify 3D objects.^{24,25} This approach consists of two main solutions: first, the one is to divide the point cloud into voxels with spatial dependencies, and then use CNN for feature extraction with each voxel as a unit. Although, this approach can preserve the spatial location information of the point cloud, it causes a large computational resource consumption due to the amount of data and the huge amount of computation that the CNN itself requires.^{26,27} Some researchers have reduced the computational demand by reducing the resolution,²⁸ but this approach has caused some loss of recognition accuracy.^{29,30} Therefore, how to transform sparse voxels into dense vectors to solve the computational efficiency problem is a main challenge for such methods.³¹ The other way is to solve the point cloud recognition task by projecting the point cloud data onto a 2D plane, using projections from different viewpoints.^{4,32} (2) Directly utilizing point cloud data for the detection and recognition of 3D objects. This approach is able to maintain the features of the original point cloud data and reduce the problems of increased workload and feature loss caused by pre-processing.^{33,34} Compared with traditional point cloud feature extraction algorithms, deep learning-based methods can more comprehensively exploit the feature information of 3D point clouds and possess broader research prospects.^{35–37} Overcoming disadvantages such as the disorderly nature of point cloud data, researchers have successfully constructed several network models for 3D object target recognition based on deep learning by directly using point cloud data as the data source.^{38–42} In PointNet,³⁸ the global feature vector is obtained by abstracting the input point cloud point by point with features and using symmetric functions to solve the problem of point cloud disorder. PointNet++³⁹ improves the problem of not being able to obtain local correlation features between points by designing a hierarchical structure on this basis, and proposes a multi-scale grouping (MSG)/multi-resolution grouping (MRG) structure that adapts to the non-uniform distribution of point cloud data and achieved a better recognition accuracy. However, these algorithms all ignore the geometric information between points, resulting in the loss of part of the local feature information, and these algorithms use the FPS algorithm for sampling, which is more dependent on the initial starting point, and different initial points will result in different sampling results and greater randomness. In recent years, researchers have studied this to some extent, but fewer studies have taken both into account.^{41–47}

To reduce the randomness of sampling while enhancing the model's ability to collect information on features, the stability of the model is enhanced. This paper proposes a minimum-bounding box over-segmentation-graph convolution 3D point cloud deep learning model, which can collect the structural information of point clouds at different scales with less sampling randomness and stronger stability. The main research contributions of this paper are: (1) a multi-scale graph convolution deep learning model is proposed, which can collect local feature information of the point cloud while enhancing the sensitivity of the model to the structure of the point cloud, enhancing the information collection capability of the model. (2) The least-enveloping box algorithm is cleverly used to reduce the randomness of sampling and increase the stability of the model. The rest of the paper is organized in the following sections. In Sec. 2, we describe in detail how the MBBOS-GCN model extracts point cloud structure information

and reduce sampling randomness. Section 3 verifies the performance of the MBBOS-GCN model using object classification and semantic scene data segmentation experiments. In Sec. 4, the stability of the model, the classification loss values and the ability to identify segmentation in complex scenes will be discussed. Finally, Sec. 5 gives the conclusions of this research.

2 Model

In this paper, we propose a minimum bounding box over-segmentation—graph convolution 3D point cloud deep learning model (MBBOS-GCN), which has the ability to collect point cloud structure information within small sampling randomness. The model uses the graph convolution module to collect structural information of point clouds and enhance the model's ability to perceive structural information (Sec. 2.1); then taking the number of sampled points as a scale to sample point clouds at different scales and collect structural information of point clouds at different scales, and a multi-scale graph convolution deep learning model is proposed (Sec. 2.2). Based on this, the minimum bounding box algorithm is utilized to reduce the sampling range, and the randomness of sampling is reduced using the FPS algorithm to sample within a small region (Sec. 2.3).

2.1 Extraction of Point Cloud Structure Information

To obtain structural information between point clouds and the spatial distribution of input points, we propose a graph convolution deep learning model. The first part of this model completes the collection of structural information of point clouds. This part first constructs a K -size neighborhood set by sampling the furthest end of the original points, treating the sampled points as nodes of the graph, and using a ball query to search for each node in the sequence by a certain radius range for the points in the domain together with the original nodes. In this K -sized neighborhood, we associate each point with the nearest point combination, collect the structural information of the point cloud by convolutional layers, and aggregate it by maximum pooling. This is shown in Fig. 1. The second part completes the collection of local feature information of the point cloud. Referring to the PointNet++ structure, the sampling points are collected through the furthest sampling algorithm; the local area is constructed using the ball query algorithm based on sampling points, and the local features of the point cloud are collected by iterating the PointNet algorithm in layers. In this layer, the multi-scale model proposed by PointNet++ is used to relieve the point cloud density inhomogeneity. The local features of the point cloud obtained from the two parts of learning are merged, and the merged features are then subjected to a maximum

Fig. 1 The deep learning model network schematic diagram.

pooling operation to obtain the global features. Using multiple fully connected layers for dimensionality reduction layer by layer, the softmax classifier is used as the loss function to derive the probability of k -class classification results.

The segmentation network part of this paper uses a point feature propagation strategy similar to the PointNet++ model.³⁹ Feature propagation is achieved by interpolating upper level features through inverse distance weighted interpolation algorithm. The features obtained after interpolation are then connected with features from the set abstraction level across chain nodes to obtain a new feature combination, which is delivered through a multi-layer perceptual vector machine similar to the PointNet.³⁸ The above process is repeated until the features are propagated to the original point set and the point cloud is finally segmented to obtain the segmentation result.

2.2 Multi-Scale Graph Convolution

When extracting structural features from point cloud, the variations of points sampled number decides directly the sampling density, which can affect the learning of structural features. When graph convolution depends on a unique sampling number, the structural features it collected will be biased toward the structural features at that number of sampling points. For example, when the sampling number is large, the structural information collected by the graph convolution will be more subtle; conversely, the structural features collected will be relatively rough, and lack some local subtle spatial information. Therefore, this paper takes the number of sampled points as the scale, sampling the point clouds at different scales for collecting the structural information of the point clouds in these scales, and increases the perception of the structural features of the point clouds by merging the structural features collected at each scale (Fig. 2) to improve the accuracy of classification segmentation.

2.3 Minimum Bounding Box Over-Segmentation Sampling

In the point cloud deep learning network model, the sampling is performed using FPS, which is susceptible to seed sampling points, resulting in a large randomness of sampling. The minimum bounding box is used for over segmentation on the point cloud. By dividing the field point cloud into many small grid blocks, a fixed size cluster of point clouds is obtained, which is as input data of FPS (shown in Fig. 3) in the small area to reduce the randomness of sampling. The minimum bounding box over-segmentation sampling uses the oriented bounding box algorithm to calculate the bounding box of the point cloud, and the bounding box is divided into several small regions of the same size according to the size of the bounding box. Then the model will estimate whether the points in each small region are empty, and the empty region will be directly deleted, on the contrary, according to the points density in the region, the number of sampling points is

Fig. 2 GCN (multiscale) deep learning model network structure diagram.

Fig. 3 MBBOS-GCN deep learning model network structure diagram.

determined based on the weighted average method, and the sampled points are combined after sampling with FPS algorithm. Finally, the sampled points are fed into the graph convolutional learning network model to form an MBBOS-GCN.

3 Results

This paper validates the performance of the model in both object classification and semantic scene segmentation. To maintain comparability with the PointNet++ deep learning model as well as other models, the ModelNet40 dataset⁴⁸ is used for the object segmentation experiments and the ScanNet dataset⁴⁹ is used for the semantic segmentation of large scenes and semantic 3D dataset⁵⁰ in the compressed version of the data reduced8 dataset. The experimental datasets are treated in the same way as the PointNet++ model experiments to ensure the fairness.

For the single-scale graph convolutional deep learning network model, various numbers of sampling points result in different spatial resolutions of objects and different spatial structure information. To ensure the reliability of the experiments, we choose the number of sampling points for a moderate sampling point density. Specifically, in the object classification experiments, 512 sampling points are selected as the input data for the graph convolution part of the single-scale graph convolution model, and in the semantic segmentation experiments, 1024 sampling points are selected as the input data for the graph convolution part of the single-scale graph convolution model. For the multiscale experiments, there is a continuous change of scale and a large spatial number of scales. As the computer equipment has limited storage and operation space, if all the scales are involved in the operation, it will cause a large amount of resources to be occupied, increase the operation burden, and even lead to the crash of the whole running program. To run the multi-scale graph convolutional deep learning network model with limited hardware equipment and to verify the effectiveness of the structure, this paper selects a limited number of scales (the smallest multi-scale, i.e., three scales) to test the performance of the model. Due to the distinction of experimental data, the scales are taken differently for the different experiments, as shown in Table 1.

In addition, in the process of training a deep learning model, the settings of the network parameters will affect the results of the experiment, and choosing the appropriate network parameters can effectively improve the performance of the model. This paper refers to the settings of the network parameters in the PointNet and PointNet++ models, which are shown in Table 2. To ensure the accuracy of the model and prevent the overfitting, this experiment draws on the results of the MSS-PointNet⁵¹ study and uses $L2$ regularization to reduce the degree of overfitting. At the same time, the size of the batch process will directly affect the accuracy of the direction of the gradient descent of the model. The larger the number of batch processes, the more accurate the direction of the gradient descent of the model and the better the effect of model convergence. As the batch process will occupy a large amount of video memory space, the more

Table 1 Setting of multi-scale parameter.

Experiment	N1	N2	N3	L1			L2			L3		
				Np1	R1	Ns1	Np2	R2	Ns2	Np3	R3	Ns3
Object classification	256	512	1024	512	.02	64	128	.04	64	None	None	None
Semantic scene data segmentation	512	1024	2048	512	.02	4	128	.04	64	None	None	None

Note: N1, N2, and N3 are the number of sample points of graph convolution; L1, L2, and L3 are MSG; NP1, NP2, and NP3 are the number of sample points of FPS; R1, R2, and R3 are sphere query radius; NS1, NS2, and NS3 are sphere query points

Table 2 Setting of network parameter.

Parameter	PointNet	PointNet++	GCN (singlescale)	GCN (multiscale)
Optimizer	Adam	Adam	Adam	Adam
Activation function	ReLu	ReLu	ReLu	ReLu
Learning rate	0.001	0.001	0.001	0.001
Dropout	0.7	0.5	0.5	0.5
Batch size	32	32	16	16
Regularization	—	—	L2	L2

intricate the batch process, the larger the video memory space consumed. Due to the limitation of the experimental hardware device, when setting the same batch_size as PointNet and PointNet++, the hardware device's video memory space is overflowed. To maintain the normal state of experiment and ensure the accuracy of the direction of gradient descent, the batch size is set to 16 in this paper. The deep learning model in this paper is built and run based on the TensorFlow deep learning framework using Ubuntu 18.04 operating system. The computer is specifically configured with an Intel Xeon Gold 6154 CPU, 32G of running memory and an NVIDIA GeForce Rtx 2080ti graphics card.

3.1 Point Set Classification in Euclidean Metric Space

We evaluated our algorithm on the ModelNet40 test set. We split the experimental data into 9843 training data and 2468 test data. A uniform sample of 1024 points on the mesh surface of the CAD model was used as the original shape of the object, and they were normalized to the unit sphere weight. During the training, we dithered the positions of the sample points by randomly scaling and rotating the object along the Z axis, adding Gaussian noise with a mean of 0 and variance of 0.02, as a means of data expansion and increasing the training point cloud. The PointNet and PointNet++ models that use point cloud data directly as input data have higher classification accuracy than models that use other types of data as input (Table 3). With the same data source, the single-scale graph convolutional deep learning model achieved a classification result of 91.12% on the ModelNet40 test set, higher than that of PointNet (at 1.92%) and PointNet++ (at 0.42%). This result shows that the structural features of the point cloud data collected by the graph convolution algorithm effectively improve the accuracy of the model. Although only the smallest three scales are used for the multi-scale model, the multi-scale graph convolutional deep learning model still reaches up to 91.57% in classification results on the ModelNet40 test set. An improvement of 0.45% indicates that increasing the number of scales sampled can improve the accuracy of the model's classification. Compared to the multi-scale graph convolution deep learning model, the classification accuracy of the minimum bounding box over-segmentation-graph convolution model improved by 0.3%, and compared to the PointNet++ model, the classification accuracy improved by 1.8%, which indicates that the

Table 3 Classification results of different models on ModelNet40 test set.

Model	Input data format	OA (%)
Subvolume	Voxels	89.2
O-CNN	Octree voxels	90.6
PointNet	Point cloud	89.2
PointwiseCNN	Point cloud	86.1
PointNet++	Point cloud	90.7
KD-NET	Point cloud	91.8
DGCNN(32)	Point cloud	92.2
DGCNN(16)	Point cloud	91.83
GCN (single scale) (ours)	Point cloud	91.12
GCN (multiscale) (ours)	Point cloud	91.57
MBBOS-GCN (ours)	Point cloud	91.87

classification accuracy of the objects can be improved by reducing the randomness of sampling by dividing the input data through the minimum bounding box. In our study, we found that the classification accuracy of the DGCNN model is higher than that of the MBBOS-GCN model. Analyzing the reason further, we found that the DGCNN batch_size is 32. Generally, the larger the batch_size, the more accurate the gradient descent direction of the model and the faster the convergence, but the video memory utilization will increase.⁵² To ensure the fairness of the experiment, we reduced the batch_size of DGCNN to 16, and after several experiments, our model was similar to the DGCNN model. To test whether there is a significant difference in the accuracy of the models, we set batch_size to 16 and conducted 30 replicate experiments on the ModelNet40 test set for O-CNN,⁵³ PointNet++,³⁹ KD-NET,⁵⁴ DGCNN,⁴⁴ and MBBOS-GCN, respectively, and by ANOVA the difference is significant at 0.03, which is <0.05 level. Our model is more stable and has better performance.

To further analyze the performance of each model, the classification accuracy of each category on the ModelNet40 test set is tallied and the results are shown in Table 4. The single-scale graph convolution model achieved 100% accuracy in category classification in six categories:

Table 4 Classification accuracy of various models for each category in the ModelNet40 test set.

Category	Type		
	GCN (single-scale)	GCN (multiscale)	MBBOS-GCN
Aircraft	1.000	1.000	1.000
Bathtub	0.920	0.924	0.934
Bed	0.980	0.970	0.970
Bench	0.750	0.780	0.800
Bookshelf	0.970	0.970	0.970
Bottle	0.960	0.955	0.972
Bowl	1.000	1.000	1.000

Table 4 (Continued).

Category	Type		MBBOS-GCN
	GCN (single-scale)	GCN (multiscale)	
Car	0.990	0.995	0.995
Chair	0.970	0.970	0.960
Vertebrae	0.950	0.960	0.960
Mug	0.800	0.850	0.890
Window curtain	0.900	0.900	0.910
Desk	0.884	0.884	0.886
Door	0.900	0.920	0.925
Dressing table	0.767	0.832	0.862
Flowerpot	0.150	0.159	0.250
Glass case	0.930	0.930	0.970
Guitar	1.000	1.000	1.000
Keyboard	1.000	1.000	1.000
Lamp	0.850	0.830	0.870
Laptop	1.000	1.000	1.000
Mantelpiece	0.980	0.970	0.970
Display	1.000	1.000	1.000
Bedside cabinet	0.980	0.965	0.955
People	0.900	0.960	0.975
Piano	0.950	0.920	0.950
Plant	0.830	0.850	0.870
Radio	0.850	0.850	0.840
Range hood	0.960	0.950	0.950
Sink	0.850	0.900	0.940
Sofa	0.980	0.997	0.997
Staircase	0.900	0.940	0.950
High stool	0.850	0.900	0.940
Table	0.840	0.840	0.870
Tent	0.950	0.950	0.950
Toilet seat	0.990	0.990	0.995
TV stand	0.860	0.830	0.840
Vase	0.810	0.810	0.830
Wardrobe	0.700	0.750	0.800
Game console	0.800	0.810	0.830

aircraft, bowl, guitar, keyboard, laptop, and display. And 36 categories' accuracy are higher than 80%, accounting for 90% of the categories in the whole dataset. Six categories (aircraft, bowl, guitar, keyboard, laptop, and display) express remained classification accuracy in 100%, and the accuracy of cars and sofas approximately up to 99.5% in multiscale graph convolution model. In addition, compared to the single-scale graph convolutional deep learning model, the number of categories with accuracy higher than 80% rises from 36 to 37. In the category classification accuracy of the minimum bounding box-graph convolution model, the accuracy of each category except flowerpot is above 80%. The classification accuracy for toilets also improved to 99.5%. Although the accuracy for flower pots is still low, there is also an improvement of nearly 10% from 15.9% in the multi-scale graph convolutional deep learning model to 25%. There are also improvements in recognition accuracy for objects such as bathtubs, benches, bottles, cups, curtains, dressers, glass boxes, lamps, people, pianos, plants, sinks, stairs, high stools, tables, toilets, vases, wardrobes, and game consoles.

3.2 Point Set Segmentation for Semantic Scene Labeling

To validate the performance of our model in large-scale point clouds, semantic scene segmentation experiments are conducted in this paper. The ScanNet dataset is applied to the graph convolutional deep learning model in this paper. To ensure fairness of the experiments, the paper kept the same treatment as 3DCNN,³⁹ PointNet, and PointNet++. The 1513 scenes in the ScanNet dataset were divided into 1201 training scenes and 312 test scenes, and 8192 points were sampled for each scene for input. Relying only on the geometry, all RGB information was removed and the point cloud labels were replaced with voxel labels. The results are shown in Fig. 4. The single-scale graph convolutional deep learning network model segmentation accuracy is 0.845, which is 0.6% higher compared to the PointNet++ (MSR) model and also higher than the accuracy of PointNet, 3DCNN. It suggests that by increasing the ability to collect structural information from point clouds, the model can be enhanced for the recognition and segmentation of complex scenes. The segmentation accuracy of the multi-scale graph convolution deep learning network model is 86.3%, which is 1.8% points higher than that of the base model PointNet++ (MSR), and also precedes that of single-scale graph convolution deep learning network model (1.2% higher) and various other models. It demonstrates that by increasing the sampling scale of the graph convolution and collecting structural information from point clouds at different scales can effectively improve the segmentation accuracy, which is important for understanding and learning complex scenes. The MBBOS-GCN segmented with 89.5% accuracy, which is a 3.2% improvement in segmentation accuracy compared to the multi-scale

Fig. 4 Segmentation accuracy of various models in semantic scenes.

Table 5 Results of outdoor large scale point cloud segmentation.

	mIoU (%)	OA (%)	Man-made	Natural	High-veg.	Low-veg.	Building	Hard-scape	Scanning-art	Cars
SnapNet ⁵⁵	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud ⁵⁶	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF_MSSF ⁵⁷	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
MSDeepVoxNet ⁵⁸	65.3	88.4	83.0	67.2	83.8	36.7	92.4	31.3	50.0	78.2
ShellNet ⁵⁹	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet ⁶⁰	70.8	91.9	86.4	77.7	88.5	60.6	94.2	37.3	43.5	77.8
SPG ⁶¹	73.2	94.0	97.4	92.6	87.9	44.0	83.2	31.0	63.5	76.2
KPConv ⁶²	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	77.3	79.7
randLA-Net ⁶³	76.0	94.4	96.5	92	85.1	50.3	95.0	41.1	68.2	79.4
MBBOS-GCN (ours)	74.9	95.6	96.6	78.7	88.8	47.9	95.8	41.9	69.2	80.3

graph convolution model and 5% improvement compared to PointNet++ (MSG). The experimental results show that the segmentation accuracy of the model can be well-improved by minimizing the randomness of sampling by over-segmenting the point cloud data with a minimum bounding box, which is important for the segmentation of semantic scenes, especially those with large data volumes.

3.3 Outdoor Site Point Cloud Segmentation Experiments

To further validate the performance of our model in outdoor large-scale point clouds, we applied the MBBOS-GCN model to a compressed version of the Semantic 3D dataset⁵⁰ in the reduced8 dataset. The results are shown in Table 5. We evaluated the segmentation results using mIoU and overall accuracy (OA). mIoU is a measure of the accuracy of the corresponding object and is used to measure the correlation between the true and predicted values, and its value is the quotient of the intersection of the true and predicted values of the object divided by the concatenation of the two; the higher the correlation, the higher the IoU value. The mIoU of the least-enveloping-box-over-segmentation-graph-convolutional deep learning model is 74.9%, which is 0.11 lower than the 76.04% of RandLA-Net, but the OA is 0.12 higher than that of RandLA-Net. In terms of the accuracy of each category, the least-enveloping-box-over-segmentation-graph-convolutional deep learning model had the highest segmentation accuracy for high veg., buildings, hard-scape, scanning-art, and CARS recognition, indicating that our model has superior performance in outdoor large-scale point clouds.

4 Discussion

4.1 Stability of the MBBOS-GCN Model

To test the stability of the model and the comparability of the experiment, this paper adopts the PointNet++ model stability test method, randomly discarding some points and using point cloud data of different densities as the input data of the model to test the stability of the model, taking the classification experiment of ModelNet40 dataset as an example, using 1024, 512, 256, and 128 points as the input data, respectively, to test. The test results are shown in Fig. 5. As the density of input points decreases, the accuracy of each model decreases as well. When the point density is >256, the single-scale graph convolution neural network, multi-scale graph convolution neural network, and PointNet++ all present good accuracy and stability with change of points number. While compared with the PointNet++ model, the stability of the model in this

Fig. 5 Robustness of various models.

paper is better. When the number of input points declines to 128, the accuracy of the single-scale and multi-scale graph convolutional neural network model is dramatically affected but still higher than that of the PointNet++ model, and the decline trend is also gently than that of the PointNet++ model. It demonstrates that the single-scale and multi-scale graph convolutional neural network model in this paper has good stability for non-uniform and sparse point data.

When the input points reduced by 50% of MBBOS-GCN model, the classification accuracy of the model was decreased by 0.29%, which is less decline than the DGCNN model as well as the multi-scale model. However, with the further declining of input points, narrowed FPS sampling range results in a decreased sampling randomness. During this time, there is less difference between MBBOS-GCN model and the multiscale graph convolution model in classification accuracy, regardless of input points number. However, compared with DGCNN, the performance of our model has clearly gained some advantages over the DGCNN model. Overall, if there are enough input points, the stability of the model can be enhanced through reduced sampling randomness, realizing by over-segmenting the input point cloud with the minimum bounding box.

4.2 Robustness of the MBBOS-GCN Model

The smaller the value of the loss function, the better the robustness of the model. Figure 6 shows the statistics of classification loss values for the single-scale graph convolution, multi-scale graph convolution, MBBOS-GCN models, and Pointnet++. In the early stage of training, PointNet++ classification loss experiences significant decrease, whereas the single-scale, multi-scale, and MBBOS-GCN models decrease gently. In the later stages of training, the single-scale, multi-scale, and minimum-wraparound-box-graph convolution models outperformed Pointnet++. By comparing the parameter settings of each model, it can be found that due to the limitation of hardware equipment, the batch processing size of single-scale, multi-scale, and MBBOS-GCN is set to be smaller than that of PointNet++, which are 16 and 32, respectively. To verify our inference, we set the PointNet++ batch size to 16, as shown in Fig. 6, and the variation of PointNet++ classification loss values slow down significantly in the pre-training period.

In terms of the trend in classification loss values change, there is a significant overfitting phenomenon as the PointNet++ model was overtrained and the training loss values rose. The single-scale, multi-scale, and MBBOS-GCN models utilizing L_2 regularization, which mitigates the overfitting phenomenon, and the fluctuation trends are generally downward.

To further analyze the performance of the graph convolutional neural network, the misclassification of some objects is statistically analyzed, and the confusion matrix of misclassification is statistically calculated for flower pots, which has the lowest accuracy rate, as shown in Table 6.

Fig. 6 Classification loss.

From the confusion matrix of flower pots, it can be seen that flower pots are more often misclassified as bottles, vases, bowls, plants, etc. Among them, misclassified flower pots as bottles are about 45.7%, the similar appearance of these two items resulting in the highest proportion of misclassification. Plant pots are incorrectly classified as plants by the program due to the plants growing in them, and the misclassified rate is about 24% of cases. Several other types of misclassified objects account for ~14.4% of the overall misclassification. By reducing the randomness of sampling through the over-segmentation algorithm, the proportion of misclassified bottles and plants with distinct differences in shape decreases from 45.7% to 39.7% and 24% to 17.9%, respectively, whereas the misclassification of bowls and vases with similar shapes do not improve significantly. It shows that the influence of the program on the random or uneven sampling can be effectively improved and the recognition accuracy of the program can be raised by reducing the randomness of the sampling by over-segmentation.

4.3 Recognition Segmentation of Objects in Complex Scenes

To further validate the recognition segmentation effect of our model, this paper applies the MBBOS-GCN model to verify the performance of the model in recognition and segmentation in complex scenes. In this paper, we use the compressed version of data reduced8 from the Semantic 3D dataset⁵⁰ as the training sample, and use the point cloud data collected by terrestrial laser scanning was used as the test sample. However, because the scene location of reduced8 is located in Europe, the architectural style differs from that of the test sample. To reduce the impact of the training samples on the experiment and for workload saving, some samples are added to the reduced8 dataset to reduce the error caused by the differences in the training samples (Fig. 7). The Gaussian filtered is used to remove the redundant noise points (which is induced mainly by manual labeling of the samples) to obtain pure samples.^{64–66} The pure point cloud is down scaled at 0.01 m to fit the format of reduced8 dataset.

To test the application capability of the model, this paper selects a more complex unfamiliar scene as the test sample—Liu Mingchuan’s former residence, and uses Topcon GLS-2000 LiDAR to scan externally to obtain the original point cloud data. Then through its supporting software ScanMaster for stitching and true color attachment, takes the same method for filtering and down scaling as adding samples before. A complex test sample of unfamiliar scenes was obtained as shown in Fig. 8.

The OA of the recognition segmentation results reaches up to 97.53%. The recognition results are shown in Fig. 9, and the recognition confusion matrix for each category is statistically analyzed in Table 7. As the test samples are located in scenic areas, no car samples are captured.

Table 6 The multiscale diagram convolved with the flowerpot classification confusion matrix.

	Aircraft	Bathtub	Bed	Bench	Bookshelf	Bottle	Bowl	Car	Chair	Vertebrae
Flowerpot	0	0	0	0	0	0.457	0.06	0	0	0
	Mug	Window curtain	Desk	Door	Dressing table	Flowerpot	Glass case	Guitar	Keyboard	Lamp
Flowerpot	0	0	0	0	0	0.159	0	0	0	0
	Laptop	Mantelpiece	Display	Bedside cabinet	People	Piano	Plant	Radio	Range hood	Sink
Flowerpot	0	0	0	0	0	0	0.24	0	0	0
	Sofa	Staircase	High stool	Table	Tent	Toilet seat	TV stand	Vase	Wardrobe	Game console
Flowerpot	0	0	0	0	0	0	0	0.084	0	0

Fig. 7 Sample. **Fig. 8** Manually label the test sample. **Fig. 9** Recognition effect.

The recognition accuracy for artificial terrain, natural landscape terrain, high vegetation, buildings, and remaining hardscape are all above 93%. The recognition accuracies for low vegetation and scanning artifacts are above 81%, which is slightly lower compared to the other categories.

To explain the reasons for the lower recognition accuracies of the low vegetation and scanning artifacts, the classified low vegetation and scanning artifacts are visualized as shown in Fig. 10. Figure 10(a) shows the mis-scored portion of the low vegetation. Due to the influence of the stitching mechanism of the sampling equipment, the scanning process requires a target on a tripod, and the point cloud scene contains a tripod point cloud. During the statistical analysis, objects such as tripods are classified as hard landscapes while the program misidentifies such objects as low vegetation during the recognition process. Figure 10(b) shows the scanning

Table 7 Identification result confusion matrix.

Actual	Prediction								
	1	2	3	4	5	6	7	8	
1	1314501331	1483810	37698	86968	45589	47899	16256	0	0.994
2	1367230	24945964	44373	414562	18152	27291	6524	0	0.93
3	45231	34520	117056401	1142681	4612	28405	54031	765	0.989
4	306594	175385	113130	6947509	7404	924176	100701	343	0.81
5	247130	146824	87441	23413	91724376	2495312	0	168	0.968
6	324572	136792	463592	1043216	3985912	83941420	14325	30312	0.933
7	53931	42831	83427	504100	1253	14431	3013312	4127	0.811
8	0	0	0	0	0	0	0	0	0

Note: 1-artificial terrain; 2-natural terrain; 3-high vegetation; 4-low vegetation; 5-building; 6- Remaining hard-scape; 7- scanning artifact; 8-car.

Fig. 10 Misclassification.

artifact misclassification. There are some scanning artifacts from the movement of the surveyor during the scanning process, but some of the movement is short and the point cloud does not move much when the scan was recorded, causing the program to misclassify this part of the point cloud as low vegetation.

4.4 Research Deficiency and Prospect

Deep learning requires a huge number of training samples. By comparing the results of the experiments with point cloud samples added in the style of some of the characteristics of the experimental site and those using the reduced8 dataset directly as training samples, we found that adding some of the point cloud samples in the style of the characteristics of the experimental site to the samples significantly improved (2.31%) the accuracy of the segmentation. During the training time, whether the samples match the characteristics of the case site directly affects the accuracy of the model. In addition, although this paper adds some point cloud samples with the style of experimental site characteristics to the public dataset reduced8 dataset in the complex scene experiment, the number of such samples is relatively small, which has limited the accuracy of the experiment.

In this paper, we use the minimum bounding box algorithm to over-segment the point cloud to reduce the size of the sampled regional space, which can effectively drop the randomness of sampling. However, this method fundamentally depends on FPS algorithm for sampling, which

still has some inevitable randomness. Furthermore, when MBBOS-GCN model over-segments the point cloud, the point cloud is divided into several small regions, where the number of regions is an empirical value (called N). And we find that the experimental accuracy can be evidently affected by adjusting Different N values. The test results show that there is a threshold in the divided regional area, and the model's classification accuracy will fall down after reaching the peak in this divided threshold, and then tend to be stable if it continues to increase empirical value N . How to make the N value adaptive to different models will be the focus of future work.

5 Conclusion

In this paper, we propose an MBBOS-GCN, which uses the minimum-bounding box algorithm to divide the point cloud into several small regions, in which the FPS algorithm is applied to reduce the randomness of sampling. We take the number of sampled points as the scale, and the improved graph convolution algorithm is used to collect structural information of the point cloud at different scales. Object classification and semantic scene segmentation experiments are conducted to validate our model:

1. The model has a higher classification accuracy. The classification segmentation accuracy reach up to 91.87% and 89.5% achieved by MBBOS-GCN on the ModelNet40 dataset and ScanNet dataset, respectively, which is higher than the other model.
2. The stability and robustness of the model are satisfied. The stability and robustness of the model are verified by comparing the fluctuation trend with varying the number of input points of the model, and the results showed that the model has reliable stability and robustness.
3. The model performs well in complex scenes. The model achieves an OA of 97.53% in the complex scene point cloud of Liu Mingchuan's former residence, which indicates that the model can be better adapted to complex scenes.
4. In this paper, we have added some point cloud samples with the characteristics of the experimental site to the public dataset reduced8, but the number of samples is relatively small, which has limited the accuracy of the experiment. Therefore, the next step of the study will be to create more samples with the style of the experimental site to meet the requirements of the experimental training sample library.

Acknowledgments

This research was funded by The National Natural Science Foundation of China (Grant No. 71603003), Youth project of Natural Science Foundation of Anhui Province (Grant No. 1908085QC1). The authors are grateful to the colleagues who participated in the data collection. Thanks go to Qu Le'an for their feedback.

References

1. C. Giuliano et al., "Computational vision and business intelligence in the beauty segment—an analysis through Instagram," *J. Marketing Manage.* **7**(2), 11–17 (2019).
2. G. R. Jose et al., "Editorial: special issue on computational intelligence for vision and robotics," *Neural Comput. Appl.* **28**(5), 853–854 (2017).
3. D. B. Tracy et al., "Computer face-matching technology using two-dimensional photographs accurately matches the facial gestalt of unrelated individuals with the same syndromic form of intellectual disability," *BMC Biotechnol.* **17**(1), 90 (2017).
4. X. Z. Chen et al., "Multi-view 3D object detection network for autonomous driving," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3478–3489 (2017).
5. F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
6. A. G. Howard et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).

7. F. Fares et al., “Wheat head detection using deep, semi-supervised and ensemble learning,” *Can. J. Rem. Sens.* **47**(2), 198–208 (2021).
8. H. S. Li et al., “A survey of 3D data analysis and understanding based on deep learning,” *Chin. J. Comput.* **43**(1), 41–63 (2020).
9. D. Mendes et al., “Design and evaluation of a novel out-of-reach selection technique for VR using iterative refinement,” *Comput. Graphics* **67**, 95–102 (2017).
10. P. Li, X. Chen, and S. Shen, “Stereo R-CNN based 3D object detection for autonomous driving,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2345–2356 (2019).
11. X. Chen et al., “Monocular 3D object detection for autonomous driving,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2147–2156 (2016).
12. B. Li et al., “GS3D: an efficient 3D object detection framework for autonomous driving,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1019–1028 (2019).
13. A. Kun, Y. Li, and J. Rehg, “3D-RCNN: instance-level 3D object reconstruction via render-and-compare,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3559–3568 (2018).
14. H. Murase and S. K. Nayar, “Visual learning and recognition of 3D objects from appearance,” *Int. J. Comput. Vis.* **14**(1), 5–24 (1995).
15. Z. Y. Qin et al., “Triangulation learning network: from monocular to stereo 3D object detection,” in *Proc. of the IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 7615–7623 (2019).
16. H. Su et al., “Multi-view convolutional neural networks for 3D shape recognition,” in *Int. Conf. Comput. Vision*, IEEE, pp. 945–953 (2016).
17. B. Shi et al., “Deep panoramic representation for 3D shape recognition,” *IEEE Signal Process. Lett.* **22**(12), 2339–2343 (2015).
18. A. Sinha, J. Bai, and K. Ramani, *Deep Learning 3D Shape Surfaces Using Geometry Images*, Springer International Publishing, Berlin (2016).
19. W. J. Chang et al., “Point clouds model recognize method based on the recognize technology of support vector machine,” *Mach. Des. Manuf.* **1**(9), 104–106 (2009).
20. O. Kahler et al., “Very high frame rate volumetric integration of depth images on mobile devices,” *IEEE Trans. Vis. Comput. Graphics* **21**(11), 1241 (2015).
21. A. Alireza et al., “3D Lidar-based static and moving obstacle detection in driving environments: an approach based on voxels and multi-region ground planes,” *Rob. Auton. Syst.* **83**, 299–311 (2016).
22. S. Song and J. Xiao, “Sliding shapes for 3D object detection in depth images,” *Lect. Notes Comput. Sci.* **8694**, 634–651 (2014).
23. D. Z. Wang and I. Posner, “Voting for voting in online point cloud object detection,” in *Rob. Sci. Syst.* (2015).
24. B. S. Kim, P. Kohli, and S. Savarese, “3D scene understanding by Voxel-CRF,” *Int. Conf. Comput. Vision*, IEEE, pp. 1425–1432 (2014).
25. Q. S. Qian et al., “Summed volume region selection based three-dimensional automatic target recognition for airborne LIDAR,” *Defence Technol.* **16**, 535–542 (2020).
26. M. Engelcke et al., “Vote3deep: fast object detection in 3d point clouds using efficient convolutional neural networks,” in *IEEE Int. Conf. Rob. and Autom.*, pp. 1355–1361 (2017).
27. S. Song and J. Xiao, “Deep sliding shapes for Amodal 3D object detection in RGB-D Images,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 808–816 (2016).
28. L. V. Dan et al., “Model-based recognition of 3D articulated target using lidar range data,” *Appl. Opt.* **54**(17), 5382–5391 (2017).
29. B. Yang, W. Luo, and R. Urtasun, “PIXOR: real-time 3d object detection from point clouds,” in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, IEEE, Salt Lake City, UT, pp. 7652–7660 (2018).
30. M. Liang et al., “Deep continuous fusion for multi-sensor 3D object detection,” *Lect. Notes Comput. Sci.* **11220**, 641–656 (2018).
31. M. C. Zhi, L. Li, and Y. H. Wu, “A new pose estimation method for non-cooperative spacecraft based on point cloud,” *Int. J. Intell. Comput. Cybern.* **12**(1), 23–41 (2019).
32. Y. Zhou and O. Tuzel, “VoxelNet: end-to-end learning for point cloud based 3D object detection,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 4490–4499 (2018).

33. J. Ku et al., “Joint 3D proposal generation and object detection from view aggregation,” in *IEEE/RSJ Int. Conf. Intell. Rob. and Syst.*, pp. 2564–2573 (2017).
34. X. D. Li, “Generic object recognition method based on stereo vision,” *J. Southeast Univ.* **43**(4), 711–716 (2017).
35. A. Golovinskiy, V. G. Kim, and T. Funkhouser, “Shape-based recognition of 3D point clouds in urban environments,” in *Int. Conf. Comput. Vision*, IEEE, pp. 2154–2161 (2010).
36. K. Jason et al., “Joint 3D proposal generation and object detection from view aggregation,” in *IEEE/RSJ Int. Conf. Intell. Rob. and Syst.*, pp. 2564–2573 (2017).
37. Y. Y. Li et al., “PointCNN: convolution on X-transformed points,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* Vol. 5, pp. 1123–1132 (2018).
38. C. R. Qi et al., “PointNet: deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
39. C. R. Qi et al., “PointNet++: deep hierarchical feature learning on point sets in a metric space,” in *Adv. Neural Inf. Process. Syst.* (2017).
40. H. Zhao et al., “PointWeb: enhancing local neighborhood features for point cloud processing,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2019).
41. Q. Xu et al., “Grid-GCN for fast and scalable point cloud learning,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2020).
42. W. Shi and R. Rajkumar, “Point-GNN: graph neural network for 3D object detection in a point cloud,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2020).
43. J. Chen et al., “Hierarchical graph network for 3D object detection on point clouds,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2020).
44. M. Simonovsky and N. Komodakis, “Dynamic edge-conditioned filters in convolutional neural networks on graphs,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3693–3702 (2017).
45. C. R. Qi et al., “Frustum PointNets for 3D object detection from RGB-D data,” in *IEEE/CVF Comput. Vision and Pattern Recognit.*, pp. 2512–2523 (2017).
46. C. Liu et al., “A fog level detection method based on image HSV color histogram,” in *Int. Conf. Prog. Inf. and Comput.*, IEEE, pp. 373–377 (2014).
47. A. Komarichev, Z. Zhong, and J. Hua, “A-CNN: annularly convolutional neural networks on point clouds,” in *IEEE/CVF Comput. Vision and Pattern Recognit.*, pp. 1134–1153 (2019).
48. Z. Wu et al., “3D ShapeNets: a deep representation for volumetric shapes,” in *Int. Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 1912–1920 (2014).
49. A. Dai et al., “ScanNet: richly-annotated 3D reconstructions of indoor scenes,” in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
50. T. Hackel et al., “Semantic3D.net: a new large-scale point cloud classification benchmark,” *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **IV-1/W1**, 91–98 (2017).
51. X. C. Wang, “Research on 3D point cloud recognition based on deep learning,” Anhui Normal University, Wuhu (2019).
52. N.S. Keskar et al., “On large-batch training for deep learning: generalization gap and sharp minima,” arXiv:1609.04836 (2016).
53. P. Wang et al., “O-CNN: Octree-based convolutional neural networks for 3D shape analysis,” *ACM Trans. Graph.* **36**(4), 1–11 (2017).
54. R. Klokov and V. Lempitsky, “Escape from cells: deep KD-networks for the recognition of 3D point cloud models,” in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 863–872 (2017).
55. A. Boulch, B. Le Saux, and N. Audebert, “Unstructured point cloud semantic labeling using deep segmentation networks,” in *Proc. Workshop 3D Object Retrieval* (2017).
56. L. Tchapmi et al., “Segcloud: semantic segmentation of 3D point clouds,” in *Int. Conf. 3D Vision* (2017).
57. H. Thomas et al., “Semantic classification of 3D point clouds with multiscale spherical neighborhoods,” in *Int. Conf. 3D Vision* (2018).
58. X. Roynard, J.-E. Deschaud, and F. Goulette, “Classification of point cloud scenes with multiscale voxel deep network,” arXiv:1804.03583 (2018).
59. Z. Zhang, B.-S. Hua, and S.-K. Yeung, “ShellNet: efficient point cloud convolutional neural networks using concentric shells statistics,” in *IEEE/CVF Int. Conf. Comput. Vision* (2019).

60. L. Wang et al., “Graph attention convolution for point cloud semantic segmentation,” in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.* (2019).
61. L. Landrieu and M. Simonovsky, “Large-scale point cloud semantic segmentation with superpoint graphs,” in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.* (2018).
62. H. Thomas et al., “KPConv: flexible and deformable convolution for point clouds,” in *IEEE/CVF Int. Conf. Comput. Vision* (2019).
63. Q. Hu et al., “RandLA-Net: efficient semantic segmentation of large-scale point clouds,” in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.* (2020).
64. J. Huang and S. You, “Point cloud labeling using 3D convolutional neural network,” in *Int. Conf. Pattern Recognit.*, IEEE (2017).
65. L. W. Lu and D. D. Liang, “Research on combining optimization of various filtering methods for point cloud data,” *J. Anhui Normal Univ.* **41**(1), 50–54 (2018).
66. X. Wu et al., “Comparison of surface reconstruction algorithms based on point cloud data,” *J. Anhui Normal Univ.* **42**(1), 46–50 (2019).

Delong Zhan received his MS degree from Anhui Normal University. His research interest is 3D data processing and visualization.

Dongdong Liang is an associate professor. He received his PhD from Wuhan University. He is the author of more than 30 journal papers and has written 3 book chapters. His current research interests include 3D data processing and visualization.

Huiyu Jin is a postgraduate student at Fudan University and Technology. Her research interests are machine learning and its application.

Xu Wu received her BS degree from Wuhan University. His research interest is big data analysis and processing.