Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 7, July 2018, pp. 1503-1512
DOI: 10.23940/ijpe.18.07.p14.15031512

3D Convolutional Neural Network for Semantic Scene Segmentation based on Unstructured Point Clouds

Rui Zhang[a,b,*], Yan Wang[c], Guangyun Li[b], Zhen Han[a], Junpeng Li[a], and Chunying Li[a]

[a]*North China University of Water Resources and Electric Power, Zhengzhou, 450045, China*
[b]*Information Engineering University, Zhengzhou, 450052, China*
[c]*Zhengzhou Institute of Technology, Zhengzhou, 450044, China*

Abstract

The use of point cloud datasets is an inevitable trend in the analysis of natural scenes. In this paper, we propose a semantic segmentation network architecture that consumes 3D point clouds directly, which can efficiently avoid mapping 3D point clouds to 2D images. Experimental results indicate strong performance that is on par with or even better than state-of-the-art methods for semantic segmentation on the Stanford semantic parsing dataset.

Keywords: point cloud; feature representation; 3D scene segmentation; deep learning; convolutional neural network

(Submitted on March 19, 2018; Revised on April 23, 2018; Accepted on June 13, 2018)

1. Introduction

Semantic scene understanding is a critical task, not only in earth science research but also in computer vision. The classes of interest include most common objects in an urban scenario: buildings, paths, vehicles, pedestrians, poles, wires, trees, and traffic signs. Because 3D point clouds not only have 3D coordinates (*X*, *Y*, *Z*) and attribute information (such as intensity) but also have the advantages of easy obtainability, high density, and high accuracy, this data type has become an inevitable trend for analysing natural scenes. Point clouds can be effectively captured with laser range sensors; for example, laser scanners and portable structure sensors [35]. Recently, laser range sensors have become popular equipment for semantic scene parsing due to their stable 3D environment perception capability in both day and night and both indoors and outdoors [18]. Substantial progress has been made with 3D point clouds in data processing and applications, e.g., automatic city planning, cultural relic repair, mapping, and navigation [28,29].

Traditionally, semantic scene segmentation algorithms are often applied directly to raw 3D laser point clouds, such as clustering [2,3,6,7,31], random sample consensus (RANSAC) [14,16], and Hough translation [5,15,17,38]. Because point clouds are unstructured, specific data structures are usually adopted before segmentation, such as Octree, k-d tree, and other mixed-index structures [8,19,25,32,33,34,37]. These existing methods show high precision. However, because of the massive characteristics of point clouds, they easily exceed the memory limits of many computers.

Recently, with the rapid development of deep learning, several image-based convolutional neural network (CNN) models have been proposed, which can assign to every pixel a refined label automatically and effectively. Compared with 2D images, 3D point clouds can provide much more information. For instance, point clouds can naturally localize the 3D coordinates of objects, which provide crucial information for subsequent tasks such as navigation or manipulation [12]. However, semantic scene parsing directly from unstructured and inhomogeneous point clouds is surprisingly difficult. In existing methods, 3D point clouds are usually projected onto a 2D surface and inputted into CNNs, which can benefit from the well-developed image-based 2D segmentation networks. However, the projection inevitably loses or distorts useful 3D spatial information [12]. Until now, few works studied deep learning directly on raw point clouds. PointNet [21] is the

* Corresponding author.
E-mail address: zhangrui@ncwu.edu.cn

pioneer in this direction for semantic scene analysis and understanding, which was just proposed in 2017. In addition, acquiring and labelling the point cloud data for different vegetation characteristics, LiDAR viewpoints, obstacle poses, etc. is an expensive and laborious process [20]. There are only four open point cloud datasets that are used in deep learning, and their formats vary greatly. If one dataset is used to train and evaluate the network and another is used for testing, substantial time must be spent pre-processing the dataset. Thus, semantic scene understanding of point clouds based on deep learning is an emerging research direction that has broad research potential and will extend deep learning's applications.

Through the above analysis, this work mainly studies the 3D convolutional neural network based on point clouds for complex semantic scene segmentation. The key contributions of our work are as follows:

- We design a novel deep convolutional neural network architecture suitable for consuming 3D point clouds. Experiments show it is superior to the existing methods in semantic segmentation results. The mean IoU is 68.63%, and the overall accuracy is 88.13% on S3DIS *Area 6*;
- The influence of the order of input points on learning performance is analysed theoretically, and the viewpoint is verified by experiments. The sorted datasets obtain the best accuracy and minimum loss;
- The network structure concatenates global and local features several times, which not only fully considers the extraction of local features of 3D objects, but also improves the efficiency of scene semantic segmentation.

The rest of this paper is organized as follows: Section 2 reviews the available 3D point cloud datasets and the literature of the previous researches on 3D CNNs. Then, detailed designs and illustrations of the proposed CNN is presented in Section 3 and the impact of input order on the learning performance is analyzed. In Section 4, through a series of experiments on a benchmark of 3D point clouds, we evaluate our proposed network architecture, visualize the segmentation results, and discuss its performance. Finally, the conclusions and future research directions are discussed in Section 5.

2. Related Work

2.1. 3D Datasets

Until now, three-dimensional datasets were very few, and their styles mainly included Computer-Aided Design (CAD) models, 3D meshes, and point clouds. Among them, point cloud datasets are scarcer: there are only four open datasets: Sydney Urban Objects Dataset [22], KITTI [9], Stanford 2D-3D-S [4], and semantic3D.net [10]. According to the application scope, point cloud datasets are divided into two categories: indoor [4] and outdoor [9,10,22]. However, according to the application purpose, they can be divided into three categories: object classification [4,10,22], part segmentation [4], and semantic segmentation [9,10,22].

Sydney Urban Objects Dataset [22] was produced by the University of Sydney, Australian Centre for Field Robotics. This dataset contains a variety of common urban road objects scanned by a Velodyne HDK-64E LiDAR. The dataset consists of 631 individual scans, which includes 25 commonly used semantic classes: vehicles, pedestrians, signs, trees, etc.

KITTI [9], produced jointly by the Karlsruhe Institute of Technology in Germany and the Toyota American Institute of Technology, is mainly used in mobile robotics and automatic driving research. This dataset contains real-world scene data from urban, rural and highway scenarios, with up to 15 vehicles and 30 pedestrians per image, and varying degrees of occlusion and truncation. The data acquisition platform is equipped with two grey-scale cameras, two colour cameras, a Velodyne 64-line 3D laser radar (HDK-64E), four optical lenses, and a GPS navigation system. The whole dataset is composed of 389 pairs of stereo images and optical flow images, 39.2 km visual ranging sequence, and 3D point clouds. However, KITTI does not contain ground-truth information for semantic segmentation. Although some researchers have manually annotated parts of the dataset to fit their needs, [1,24,26] only generated ground-truth data for images; although [36] annotated both images and point clouds, the code and dataset are not publicly available.

Stanford 2D-3D-S [4] is a large-scale indoor spatial dataset proposed by Stanford. It provides multiple modalities: 2D (RGB image), 2.5D (depth information, surface normal vector), and 3D (grid, point cloud), all with pixel level and point level semantic annotation information. S3 DIS is a subset of Stanford 2D-3D-S, which only contains 3D point cloud data. The dataset used in this paper is S3DIS.

Semantic3d.net [10] is a large-scale 3D point cloud natural scene dataset, which was released by the *Hackel* team at the Zurich Institute of Technology, Switzerland in 2017. This benchmark closes the gap and provides a large labelled 3D point cloud dataset of natural scenes with over 4 billion points in total. It also covers a range of diverse urban scenes: churches, streets, railroad tracks, squares, villages, soccer fields, and castles.

2.2. 3D CNNs based on Point Clouds

Because of the scarcity of 3D point cloud datasets, there are fewer CNNs that are based on 3D point clouds. There are six famous CNNs: BuildingParser [4], Huang J-3DCNN [11], Boli-3DFCN [12], PointNet [21], PointNet++ [23], and PointCNN [13]. In general, the organization that proposes a 3D point cloud dataset is the same organization studying the convolutional neural network that is based on the 3D point cloud dataset, for example, Stanford University. The first two models were not publicly available. The Boli-3DFCN model was available, but it was used for object detection, not semantic segmentation.

Among the six CNNs, the last three are applicable for semantic scene segmentation. PointNet and PointNet ++ were proposed in 2017, while PointCNN was proposed in January 2018. The research on convolutional neural networks that are based on 3D point clouds is just beginning.

PointNet [21] is a general framework for object classification, part segmentation, and scene segmentation. Max-pooling layer was used as a symmetric function to deal with the unordered point clouds. Two T-net networks were used to deal with the rotation invariance of the model. The shortcoming of the model is that only one max-pooling layer was used to integrate the single point features, and the network has insufficient ability to extract local information. In order to solve this problem, PointNet++, an improved version, was proposed. PointNet++ [23] sampled the point cloud and divided the region into several parts, and then the basic PointNet network was used to extract the feature in each small area by iterating several times according to the requirement, and then fusing the global and local features of point clouds. The core of PointNet++ lies in the selection of centroid points in each region and region division method. With regard to the selection of the centroid point, a sampling method named farthest point sampling (FPS) algorithm was adopted. Multi-scale grouping (MSG) and multi-resolution grouping (MRG) were proposed for region division. The MSG approach was computationally expensive. Although the speed of MRG was greatly improved, because of the iterative use of the basic PointNet structure in the segmentation process, the computational efficiency was still lower than that of PointNet. For example, PointNet cost 11.6s in the forward propagation while PointNet++ cost 87.0s. For PointNet ++, only part of the code is available, and in the corresponding article, only the results of the classification and part segmentation are provided, not those of semantic scene segmentation. PointCNN [13] is an extension of CNN. To address the irregularity and disorder of point cloud, an x-*conv* transform was proposed. The model has a good learning effect in many kinds of shape analysis tasks, but it is far less effective in general images (such as CIFAR 10) than CNN. In addition, there is a phenomenon of over-fitting for some small-scale data.

Through the above analysis, inspired by PointNet, we design a novel deep net architecture, which directly uses raw point clouds as input data.

3. Method

3.1. Point Cloud Feature Representation

How can we convert point clouds into an understandable format of CNN? 2D images are structured, most of them are expressed in dense arrays, and the pixels are arranged in equal distances, so the 2D convolution operations can obtain a uniform output. The most direct way of extending 2D CNN to 3D data is to use 3D voxel for representing 3D data; thus, 3D CNN can be used. However, 3D data is usually sparse, but it is difficult for voxel-based 3D CNN to take advantage of this feature. Although point clouds can express 3D sparse data, the points are inhomogeneously unordered. CNN cannot directly use original point clouds to segment scenes. Thus, it is necessary to express point clouds as an understandable form of network model.

Because 3D point clouds are unstructured and inhomogeneous, the 3D coordinates of points are indispensable. In addition to the 3D coordinate, we also use the colour feature. Since points are split by room and each room is sampled into blocks of area 1m by 1m, the location relative to the room (from 0 to 1) should also be included. In summary, each point is represented by a 9-dimensional vector $[X_i, Y_i, Z_i, R_i, G_i, B_i, X^l_i, Y^l_i, Z^l_i]$, which is composed of the 3D coordinate, the RGB information, and the location of the i^{th} point in the l^{th} room. The feature representation not only clearly expresses the 3D coordinate and spectral information of each point, but also shows the spatially-local correlation.

3.2. 3D Convolutional Network Architecture

Inspired by PointNet, we propose a CNN for point clouds, which is named PC-CNN. The network architecture is visualized in Figure 1, and the detail of PC-CNN is listed in Table 1. To solve the problem of insufficient local feature extraction of PointNet and the expensive computational cost of PointNet++, the model proposed in this paper improves the segmentation

effect by increasing the network layers, adding max-pooling, and concatenating local and global features multiple times. PC-CNN has 10 convolutional layers, 1 max-pooling layer, 2 fully connected layers, and 1 *concat* layer. The local features of the 6th convolutional layer are fused with the global features of the 2nd fully connected layer.

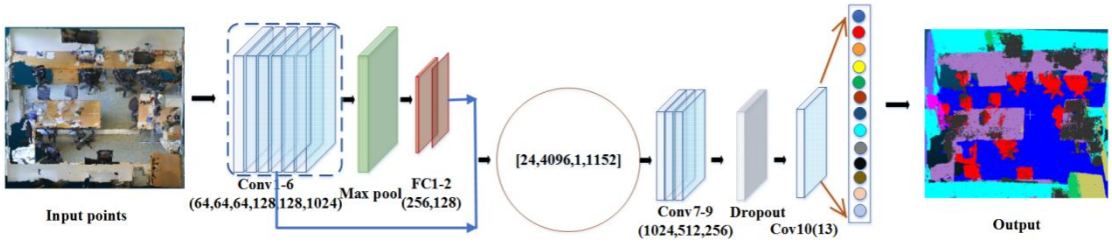


Figure 1. Illustration of the PC-CNN architecture. The circle means linking a local feature to a global feature, and the numbers in brackets at the bottom of each layer are the filter size of that layer. Batchnorm and ReLU are used for all layers.

Table 1. Details of PC-CNN.

Type	Input_shape	Filter_num	Filter_shape	Stride_shape	Padding	Output
Conv1	[24,4096,9,1]	64	[3,3,1,64]	[1,1,3,1]	VALID	[24,4096,3,64]
Conv2-Conv3	[24,4096,3,64]	64	[3,3,64,64]	[1,1,1,1]	SAME	[24,4096,3,64]
Conv4	[24,4096,3,64]	128	[3,3,64,128]	[1,1,1,1]	SAME	[24,4096,3,128]
Conv5	[24,4096,3,128]	128	[3,3,128,128]	[1,1,1,1]	SAME	[24,4096,3,128]
Conv6	[24,4096,3,128]	1024	[3,3,128,1024]	[1,1,3,1]	SAME	[24,4096,1,1024]
Max pool	[24,4096,1,1024]	--	[1,4096,1,1]	[1,1,1,1]	--	[24,1,1,1024]
Reshape	[24,1,1,1024]	--	--	--	--	[24,1024]
FC1	[24,1024]	--	--	--	--	[24,256]
FC2	[24,256]	--	--	--	--	[24,128]
Reshape	[24,128]					[24,1,1,128]
Concat (Conv6, FC2)	[24,4096,1,1024] [24,1,1,128]	--	--	--	--	[24,4096,1,1152]
Conv7	[24,4096,1,1152]	1024	[3,1,1152,1024]	[1,1,1,1]	SAME	[24,4096,1,1024]
Conv8	[24,4096,1,1024]	512	[3,1,1024,512]	[1,1,1,1]	SAME	[24,4096,1,512]
Conv9	[24,4096,1,512]	256	[3,1,512,256]	[1,1,1,1]	SAME	[24,4096,1,256]
Dropout	[24,4096,1,256]	--	--	--	--	[24,4096,1,256]
Conv10	[24,4096,1,256]	13	[3,1,256,13]	[1,1,1,1]	SAME	[24,4096,1,13]
Squeeze	[24,4096,1,13]	--	--	--	--	[24,4096,13]

In the first six convolutional layers of PC-CNN, *filter_size*=3*3, *padding*='SAME' except for the first layer, vector *stride*= [1,1,3,1] or [1,1,1,1], and the numbers of *filters* are 64, 64, 64, 128, 128 and 1024, respectively. In the last four convolutional layers, *filter_size* = 3*1. The input of every layer is a tensor $[i_1,\cdots,i_k]$ (k=4), where i_1 is the batch size, i_2 is the input height of the points, i_3 is the input width, and i_4 is the number of input channels. The output from each layer also forms a tensor $[f_1,\cdots,f_k]$ (k=4), where f_1 is the batch size, f_2 is the input height of the points, f_3 is the filter height, and f_4 is the number of output channels.

The output of the pooling layer is a global signature of the input set. To learn both global and local point features, we concatenate the local (the output of the sixth convolutional layer) and the global features (the output of the second fully connected layer) for each point, and the aggregation result is shown in the circle in Figure 1.

3.3. Impact of Input Order on the Learning Performance

To make a model invariant to input permutations, [21] introduced three strategies, namely, sorting the input into a canonical order, treating the input as a sequence, and using a simple symmetric function to aggregate the information from each point, of which sorting is a simple solution. According to PointNet, in high-dimensional space, there does not exist an ordering that is stable with respect to point perturbations in the general sense. However, in "OrderMatters" [30], the authors showed that order does matter and cannot be completely omitted.

A series of methods based on CNN have achieved great success in image recognition. The key reason is that CNN can capture the spatial local features well. Mathematically, convolutional operations in CNN are essentially a weighted summation of inputs, the result of which depends on the order of input, that is, s(a,b) is usually not equal to s(b,a), shown as Equation (1).

$$\begin{aligned} s(a,b) &= (w*x)(a,b) = Conv([w_m, w_n],[x_a, x_b]^T) \\ s(b,a) &= (w*x)(b,a) = Conv([w_m, w_n],[x_b, x_a]^T) \end{aligned} \tag{1}$$

In particular, if *a* is replaced by *e*, and the order of *e* is after *d*, the result of s(b,c,d,e) is usually quite different from that of s(a,b,c,d), shown as Equation (2).

$$\begin{aligned} s(a,b,c,\mathrm{d}) &= (w*x)(a,b,c,\mathrm{d}) = Conv([w_m, w_n, w_o, w_p],[x_a, x_b, x_c, x_d]^T) \\ s(b,c,d,e) &= (w*x)(b,c,d,e) = Conv([w_m, w_n, w_o, w_p],[x_b, x_c, x_d, x_e]^T) \end{aligned} \tag{2}$$

Therefore, convolutional operations in CNN are sensitive to the order of data input, and it is difficult to extract valid features for unordered data. Because of the order of the convolutional operation itself, but the point cloud has the characteristic of disorder, the x-*conv* operation is put forward by using KNN to select the adjacent points for convolution, and the coordinate information of the points is added to the features as a part of it [13].

If the logically adjacent points are also adjacent in the physical storage space, that is, to sort the point clouds according to a certain attribute and make them orderly under certain rules, the ability of the network to extract the local features of the 3D model can be improved. The order can be independent of the input data and rely only on coding rules, or input points can be sorted according to one property (e.g., 3D coordinates). We propose two strategies: (1) pre-processing the input data by sorting the points according to their 3D coordinates; (2) building a KD data structure that alters the order of unstructured point clouds, as detailed in [37]. In Section 4, the experimental results for an unsorted point set, a sorted point set, and a KD-tree-organized point set will be compared in terms of mean loss and accuracy. The experiments show that the order of the input data has an impact on the learning performance.

4. Experiments

To assess the results more quantitatively, we used the S3DIS semantic parsing benchmark dataset, which contains 3D scans from 'Matterport' scanners in 6 areas that include 271 rooms, and each point is annotated with one of the semantic labels from 13 categories: ceiling, floor, wall, beam, column, window, door, table, chair, sofa, bookcase, board, and clutter.

We used the *adam* optimizer with initial *learning_rate* = 0.001, *momentum* = 0.9 and *batch_size* = 24, *decay_rate* = 0.5, *decay_step* = 300000, and *max_epoch* = 51. Training on 'Area1-Area5' of the S3DIS dataset, PC-CNN took 11 hours, to converge with Tensorflow and a single INVIDIA TitanX 12 G GPU, while PointNet took half a day with a GTX1080 GPU and PointNet++ spent more than 23 hours.

This section provides qualitative results for our PC-CNN model. In Sec 4.1, we will compare our model with PointNet and PointCNN on S3DIS *Area 6*, and the comparison of two sorting methods will be carried out on the training set. Table 4 provides the per-category mean IoU. Then, Sec 4.2 illustrates the performance and Sec 4.3 presents a further analysis of our model.

4.1. Validating the Network Architecture

Generating large-scale 3D point cloud datasets for semantic segmentation is costly and difficult. Few deep neural networks can process this type of data directly. The usual practice is to randomly sample point clouds to achieve a sparse set. Then, a sparse set of key points is used for training, validation, and testing. In our experiments, 4096 points are randomly sampled in each data block for model training, while all points of '*Area6*' are used for testing.

First, we compare our two models with PointNet on S3DIS *Area 6*. The evaluation criteria that we used are mean Intersection over Union (mIoU) and overall accuracy. Comparison results are shown in Table 2. Using the reference implementation of PointNet, we reproduced the results reported by [21], as seen in the top row of Table 2. Our model significantly outperforms PointNet. mIoU and overall accuracy are increased by approximately 4.04% and 2.12%, respectively.

Table 2. Semantic segmentation results of different models on the S3DIS Area 6

Model	mIoU (%)	Overall accuracy (%)
PointNet [24]	64.01	86.01
PC-CNN	68.63	88.01

To evaluate the impact of input order on the learning performance, we performed many experiments. First, we trained the PC-CNN model with original input points, namely, unsorted points. Second, we built a KD tree to organize the 3D point clouds, and the points were stored according to the order of the KD tree leaf nodes. Then, we used the pre-processed set to train the original PC-CNN again. The results are shown in the second row of Table 3. The effect is not obvious; there is a slight improvement in accuracy. Finally, the 3D point clouds were sorted along the *X*-, *Y*-, and *Z*-axes and the model was retrained with the sorted point cloud. One point to note: the sorting speed of each file was very fast because the original training data were split by room. The results show that the mean loss is reduced by 3% and the accuracy improved by 1.2%.

Table 3. Semantic segmentation results for different input orders on the S3DIS training set

Input order	Mean loss (%)	Accuracy (%)
Unsorted	8.9840	96.6803
KD tree	8.8964	96.7164
Sorted	**5.9525**	**97.8237**

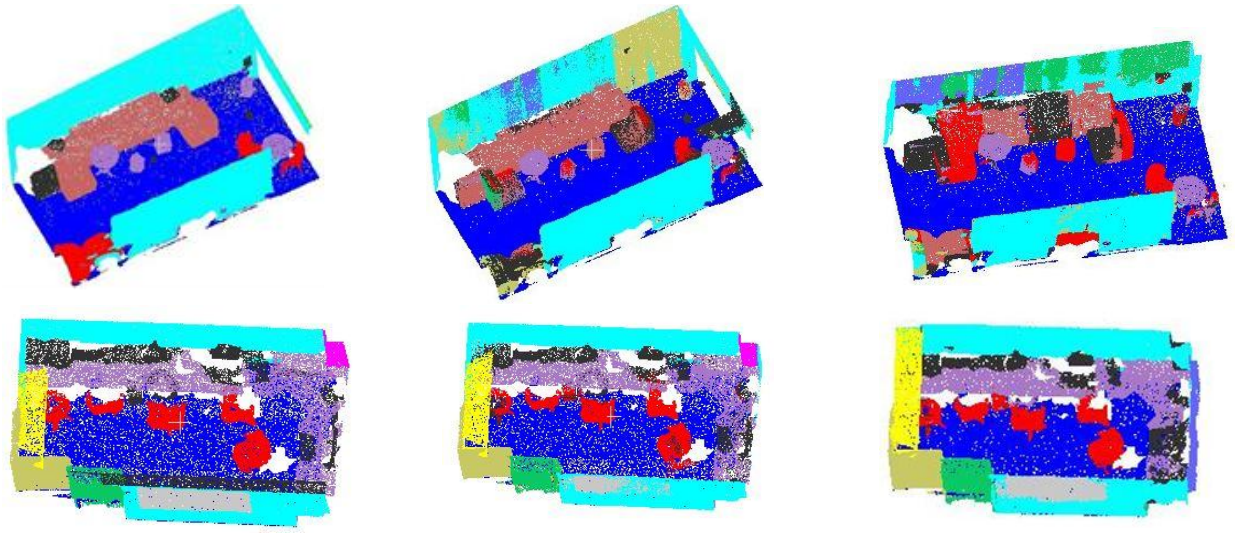
To further evaluate the performance of our model, we compare the per-category mean IoU scores with those of PointNet, which are listed in Table 4. The first row shows the segmentation results of PointNet, and the second row contains the results of our PC-CNN model. The results show that the mean IoU values of the 11 categories are improved, especially for column, table, sofa, bookcase, and board, e.g., 'column' receives an 11.47% performance boost.

Table 4. Per-category semantic segmentation results. The metric is mean IoU (%); ceil: ceiling, col: column, win: window, book: bookcase, clut: clutter.

Method	mean	ceil	floor	wall	beam	col	win	door	table	chair	sofa	book	board	clut
PointNet	64.01	91.60	97.45	73.23	63.28	40.90	**69.42**	79.32	66.99	65.19	22.46	57.76	50.30	54.17
Ours	68.63	**93.09**	**97.52**	**78.23**	**64.66**	**52.37**	69.29	**80.86**	**71.51**	**68.59**	**36.01**	**64.33**	**57.74**	**58.02**

4.2. Visualization Results

Compared with PointNet, PC-CNN can better obtain geometric features of different sizes, which is very important for understanding the multi-level scene and labelling objects of different sizes. The visualization of example scene segmentation results is shown in Figure 2. The first row shows that the segmentation effects of the PC-CNN model on sofas, chairs, and walls are obviously better than those of PointNet. It can be seen that the PointNet model misjudges parts of sofas as chairs, and the recognition effect of the wall surface is also very poor. In the second row, PC-CNN performs better than PointNet on the right side of the table.



(a) Label (b) PC-CNN (c) PointNet

Figure 2. Comparisons of segmentation effects between PC-CNN and PointNet.

Next, Figure 3 shows the panoramic view of the S3DIS *test* set, which contains 1 conference room, 1 copy room, 6 hallways, 1 lounge, 37 offices, 1 open space, and 1 pantry. Figure 5 shows the segmentation details for four types of objects: the conference room, the lounge, the office, and the open space.

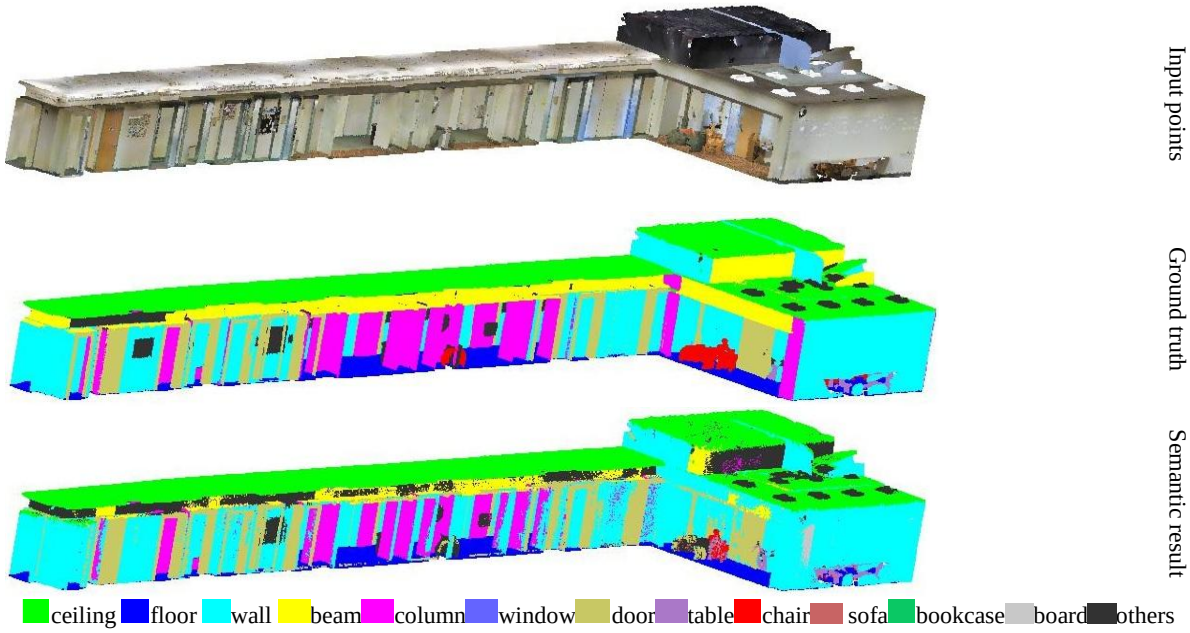


Figure 3. Panoramic view of S3DIS test set.

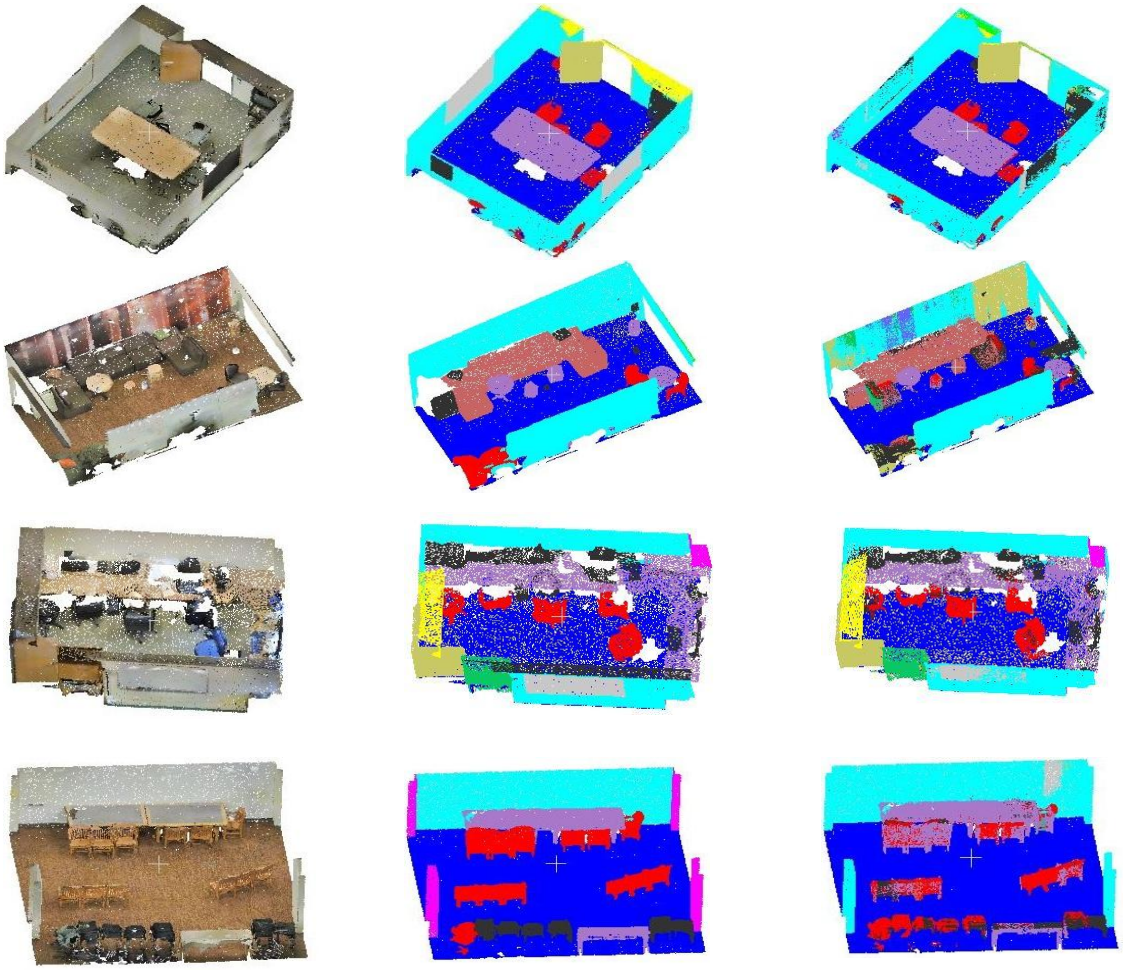


Figure 4. Segmentation details of the four types of objects. The top row: conference room; the second row: lounge; the third row: office; the bottom row: open space.

In Figure 4, the ceilings and parts of the walls are removed to show the effects of the interior. The segmentation results that are shown in Figures 3 and 4 indicate that most categories have good segmentation effects, such as floor, table, chair, sofa, door, wall, ceiling, and bookcase. The segmentation results demonstrate the stability and efficiency of our model.

4.3. Discussion

Compared with PointNet, the deep convolutional neural network model that we proposed performs significantly better. A comparison in terms of accuracy and loss during training is shown in Figure 5, where 'blue' denotes the training process of PC-CNN and 'red' denotes the training process of the PointNet model. First, the number of convolutional layers in the PC-CNN model is increased, which helps better extract the structural features of the 3D point cloud. Second, for 2D CNNs, smaller kernels help improve the efficiency of learning [27]. However, this feature is not suitable for 3D point clouds [23]. Our experiments confirmed this. For the PC-CNN model, we set *filter_size*=3*3, while *filter_size*=1*1 was used in the PointNet model. Third, we set hyper-parameter *padding* = 'SAME' for all convolutional layers except for the first one, through which we added zeros around point clouds to better segment boundary points. Lastly, we changed the input order of the point clouds, which also improved the segmentation effect to a certain extent.

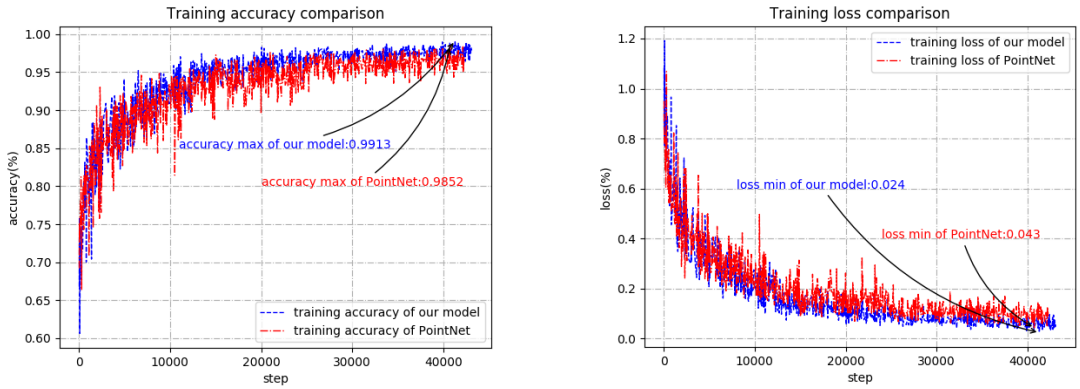


Figure 5. Training accuracy and loss comparisons between our PC-CNN and PointNet

5. Conclusions

Based on deep learning, a novel 3D CNN for point clouds is proposed in this paper. A series of experiments are performed to evaluate our model, and the segmentation results are visualized. The results show that our model outperforms state-of-the-art methods for 3D scene semantic segmentation on the S3DIS benchmark. The study of 3D convolutional neural networks with raw 3D point clouds is still in its infancy. There are still many difficult problems that need to be solved. We would like to explore several directions in future work: (1) the surface normal and curvature of points are essential properties of point clouds, and in the next study, we should try to take them into account to find a better feature representation method; (2) the impact of input order on learning performance needs to be further studied; (3) the need for an openly available 3D point cloud dataset is urgent, and the development of a 3D point cloud annotation system is a very meaningful research direction; (4) the dataset that we used is an indoor scene dataset, and the performance of the model for outdoor scenes needs to be evaluated; and (5) the input point sets were pre-processed to the same scale in our experiments, so multi-scale feature extraction needs to be further studied.

Acknowledgements

This study was undertaken with financial support from the National Natural Science Foundation of China (NSFC) (Grant no. 41501491 and 61601184) and the Key Science Research Program of Higher Education of Henan Province, China (No. 16A520062).

References

1. J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road Scene Segmentation from a Single Image," *European Conference on Computer Vision. Springer*, pp. 376–389, 2012.
2. N. H. Arachchige, H. G. Maas, "Automatic Building Facade Detection in Mobile Laser Scanner Point Clouds," *In. The German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF)*, Potsdam, Germany.2012.

3. N. H. Arachchige, S. N. Perera, H. G. Maas. "Automatic Processing of Mobile Laser Scanner Point Clouds for Building Facade Detection," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B5:187-192, 2012.
4. I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D Semantic Data for Indoor Scene Understanding," ArXiv e-prints, Feb. 2017.
5. X. Y. Ai, L. Y. Wang. "Extraction of Planar Characteristics of Airborne LiDAR Point Cloud Data," *Journal of Liaoning Technical University:Natural Science*,34(2):212-216, 2015.
6. J. M. Biosca, J. L. Lerma. "Unsupervised Robust Planar Segmentation of Terrestrial Laser Scanner Point Clouds based on Fuzzy Clustering Method," *ISPRS Journal of Photogrammetry & Remote Sensing*, 63:84-98, 2008.
7. J. P. Burochin, B. Vallet, M. Bredif, et al. "Detecting Blind Building Facades from Highly Overlapping Wide Angle Aerial Imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*. 96:193-209, 2014.
8. C. Chen, W. Ke, X. U. Wenxue, et al. "Real-Time Visualizing of Massive Vehicle-Borne Laser Scanning Point Clouds," *Geomatics & Information Science of Wuhan University*, 40(9):1163-1168, 2015.
9. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The Kitti Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
10. T. Hackel, N. Savinov, L. Ladicky and Jan D. Wegner and K. Schindler and M. Pollefeys, "SEMANTIC3D.NET: A New Large-Scale Point Cloud Classification Benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1-W1: 91-98, 2017.
11. J. Huang, S. You, "Point Cloud Labeling Using 3D Convolutional Neural Network," *International Conference on Pattern Recognition. IEEE*, 2017.
12. B. Li. "3D Fully Convolutional Network for Vehicle Detection in Point Cloud," *Computer Vision and Pattern Recognition*. 2017. arXiv:1611.08069 23
13. Y. Y. Li, R. Bu, M. C. Sun, B. Q. Chen. "PointCNN," https://arxiv.org /abs/1801.07791, 2018.
14. M. D. Li, S. P. Jiang, H. P. Wang. "A RANSCA-Based Stable Plane Fitting Method of Point Clouds," *Science of Surveying and Mapping*. 40(1), pp:102 – 106, 2015.
15. M. L. Li, G. Y. Lin, L. Wang, et al. "Automatic Feature Detecting from Point Clouds Using 3D Hough Transform," *Bulletin of Surveying and Mapping*, (2):29-33, 2015.
16. N. Li, Y. W. Ma, Y. Tang and S. L. Gao. "Segmentation of Building Facade Point Cloud Using RANSAC," *Science of Surveying and Mapping*. 36(5): 144-146. 2011.
17. Y. N. Lin, W. Wei. "Research on Algorithm of Object Tracking based on Generalized Hough Transform," *ZheJiang University*, 2013.
18. Y. Liu, F. Wang, A. M. Dobaie, et al., "Comparison of 2D Image Models in Segmentation Performance for 3D Laser Point Clouds," *Neurocomputing*, 251, 2017.
19. D. Meagher, "Geometric Modeling Using Octree Encoding," *Computer Graphics and Image Processing*, 19(2): 129-147, 1982.
20. D. Maturana, S. Scherer. "3D Convolutional Neural Networks for Landing Zone Detection from LiDAR," *IEEE International Conference on Robotics and Automation*. IEEE, pp:3471-3478, 2015.
21. C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation," arXiv preprint arXiv:1612.00593, 2016.
22. A. Quadros, J. Underwood, and B. Douillard, "An Occlusion-Aware Feature for Range Images," *Robotics and Automation, ICRA'12. IEEE International Conference on*, May pp:14-18 2012.
23. C. R. Qi, L. Yi., H. Su, & Guibas, L. J. "Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," 2017 arXiv:1706.02413v1
24. G. Ros, J. M. Alvarez. "Unsupervised Image Transformation for Outdoor Semantic Labeling," *Intelligent Vehicles Symposium*. *IEEE*, pp:537-542, 2015.
25. R. Richter, M. Behrens, J. Döllner. "Object Class Segmentation of Massive 3D Point Clouds of Urban Areas Using Point Cloud Topology," *International Journal of Remote Sensing*. Vol. 34, No. 23, 8408–8424, 2013.
26. G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. M. Lopez, "Vision-Based Offline-Online Perception Paradigm for Autonomous Driving," *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE*, pp. 231- 238, 2015.
27. K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
28. B. Song, Z. Wang and L. Sheng, "A New Generic Algorithm Approach to Smooth Path Planning for Mobile Robots," *Assembly Automation*, Vol. 36, No. 2, Apr. 2016, pp. 138-145.
29. B. Song, Z. Wang, L. Zou, "On Global Smooth Path Planning for Mobile Robots using a Novel Multimodal Delayed PSO Algorithm," *Cognitive Computation*, 9(1):1-13, 2017.
30. O. Vinyals, S. Bengio, and M. Kudlur. "Order matters: Sequence to Sequence for Sets," arXiv preprint arXiv:1511.06391, 2015.
31. P. Vracar, I. Kononenko, and M. Robnik-Sikonja. "Obtaining Structural Descriptions of Building Facades," *Computer Science and Information Systems* 13(1):23–43, 2015.
32. Y. M. Wang, M. Guo. "A Combined 2D and 3D Spatial Indexing of Very Large Point-cloud Data Sets," *Acta Geodaetica et Cartographica Sinica*,41(4):605-612, 2012.
33. J. S. Yang. "A Method of Combing the Model of the Global Quadtree Index with Local KD-tree for Massive Airborne LiDAR Point Cloud Data Organization," *Geomatics and Information Science of Wuhan University*,39(8):918-922, 2014.
34. J. S. Yang, X. F. Huang. "A Hybrid Spatial Index for Massive Point Cloud Data Management and Visualization," *Transaction in GIS*, 18(S1):97-108, 2014.

35. B. S. Yang, F. X. Liang, R. G. Huang. Progress, "Challenges and Perspectives of 3D LiDAR Point Cloud Processing," *Acta Geodaetic et Cartographica Sinica*, 46(10):1509-1516.DOI: 10.11947/j.AGCS.2017.20170351, 2017.
36. R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor Fusion for Semantic Segmentation of Urban Scenes," *Robotics and Automation (ICRA), IEEE International Conference on. IEEE,* pp. 1850-1857, 2015.
37. R. Zhang, G. Y. Li, L. Wang, M. L. Li et al., "New Method of Hybrid Index for Mobile LiDAR Point Cloud Data," *Geomatics and Information Science of Wuhan University*, 2017. DOI: 10.13203/j.whugis20160441
38. D. Y. Zhang, W. Q. Wu, M. P. Wu, et al. "Plane Landmark Detection from Lidar Data Based on3D Hough Transform," *Journal of National University of Defense Technology*, 32(2):130-134, 2010.

Rui Zhang received her MSc degree from Southwest University in 2006. She is currently a Ph.D. candidate at Information Engineering University. She works at North China University of Water Resources and Electric Power as an associate professor. Her main research interests include data processing of point clouds, computer vision, and artificial intelligence.

Yan Wang works at Zhengzhou Institute of Technology as an associate professor. Her main research interests include computer vision, social network, and network security.

Guangyun Li received his MSc degree of engineering in 1987 from Zhengzhou Institute of Surveying and Mapping. Currently, he works at Information Engineering University as a professor and doctorial supervisor. His main research interests include precise engineering and industry measurement, navigation and location services, and applications.

Zhen Han, Junpeng Li and Chunying Li are students at North China University of Water Resources and Electric Power. Their tutor is Rui Zhang.