DNNSP: A Deep Neural Network with Spatial Pooling on Point Clouds for 3D Classification

Zhen Wang[1], Liqiang Zhang[1], Tian Fang[2], Liang Zhang[1], Huiqian Ding[1]

[1]State Key Laboratory of Remote Sensing Science, Beijing Normal University,
Beijing 100875, China

[2]Hong Kong University of Science and Technology, Clear Water Bay, Kowloon,
Hong Kong, China

Abstract-The large number of object categories and many overlapping or closely neighboring objects in large-scale urban scenes pose great challenges in point cloud classification. Most works in deep learning have achieved a great success on regular input representations, but not much work has been done in deep learning on point clouds due to the irregularity and inhomogeneity of the data. In this paper, a deep neural network with spatial pooling (DNNSP) is proposed to classify large-scale point clouds without rasterization. The DNNSP first obtains the point-based feature representation. Then the distance minimum spanning tree (DMst)-based pooling is applied in the point feature representation process to recognize and describe the spatial information among the points in the point clusters. The body points and marginal points in the DNNSP are handled separately by configuring different weights for them in the feature representation. In this way, the DNNSP can learn the feature representations of points scaled from the entire regions to the centers of the point clusters, which makes the point cluster-based feature representations robust and discriminative. The proposed approach achieves high classification performance on different types of point clouds and significantly outperforms other methods. we handle

Keywords—Point cloud classification, deep neural network, spatial pooling.

1. Introduction

With the rapid advances in laser scanning technology, classification of point clouds of large-scale urban scenes efficiently and accurately is of major importance in the remote sensing and computer vision fields (Yang et al., 2015b). To classify the input point clouds, most existing approaches (Frome et al., 2004; Li et al., 2016; Weinmann et al., 2015; Zhang et al., 2013; Wang et al., 2015; Xiong et al., 2011; Yang et al., 2015a; Zhang et al., 2016; Xu et al., 2017; Zheng et al., 2017) utilize hand-crafted features for each modality independently and combine them in a heuristic manner. These approaches fail to adequately utilize the consistent and complementary information among features, which are difficult to capture high-level semantic structures. Although the features learned from most of the current deep learning methods (Bengio et al., 2013; Fukano and Masuda, 2015; Guo et al., 2015; Kragh et al., 2015) can generate high-quality image classification results, these methods do not adequately recognize fine-grained patterns to complex scenes due to the unorganized distribution and uneven point density of the data.

In contrast to images whose spatial relationships among pixels can be captured by sliding windows, the points in a point cloud are unorganized, and the point density is uneven. Through rasterizing the 3D point cloud, spatial relationships and correlations among points can be recognized. Then, deep learning technology is fit to the rasterized point cloud or 3D models (Maturana and Scherer, 2015a, b; Wu et al., 2015; Zhu et al., 2014). On the one hand, such methods work well for dense and even point clouds, but they have limitations for large-scale urban scenes in which rasterization is difficult to design for all of the objects, given the uneven point densities and missing data. The potential of deep learning techniques for large-scale point cloud classification is still relatively unexplored. On the other hand, the rasterization process also loses a large amount of valuable information about the shape and geometric layout of the objects.

With the original 3D point cloud data, we can more precisely determine the shape, size and geometric orientation of the objects (Koppula et al., 2011). Moreover, augmenting spatial cues with 3D information can enhance the object detection in cluttered, real world environments (Golovinskiy et al., 2009). In this paper, a deep neural network with spatial pooling (DNNSP) that exploits the rich relational information of the points is proposed for large-scale point cloud classification. The DNNSP can handle the raw point cloud without rasterization. Experimental results on various point clouds demonstrate that our approach outperforms other methods. The spatial pooling layers in the deep neural network significantly boost the classification performance.

2. Related Work

Many recent methods have utilized features such as spin images (Johnson and

Hebert, 1999), eigenvalues, shape and geometry features (Fukano and Masuda, 2015; Pu et al., 2011) for point cloud classification. Chehata et al. (2009) classify point clouds by using random forests with 21 features that can be categorized into five categories. Guo et al. (2015) utilize JointBoost with 26 features to classify outdoor point clouds into five classes, such as buildings, vegetation, grounds, electric wires and pylons. Kragh et al. (2015) use an SVM classifier with 13 features to classify point clouds. Brodu and Lague (2012) extract multi-scale features from different neighborhoods for classifying vegetation, rocks, water and grounds. Zhang et al. (2013) cluster point clouds by using a region growing algorithm and then use the SVM classifier with features of geometry, echoes, radiation degrees and topology of the clusters for point cloud classification. Zhang et al. (2015) utilize the Conditional Random Field (CRF) for scene semantic segmentation by fusing point clouds with images. Niemeyer et al. (2016) present a two-layer CRF that can incorporate the context with different scales. The used or obtained features in the above methods are sensitive to local geometric noise, and they do not adequately capture the global structure of the shape (Xie et al., 2015).

The deep learning can automatically jointly learn features and classifiers from the data (Stuhlsatz et al., 2012) and has shown flexibility and capability in many applications, such as image classification (Krizhevsky et al., 2012), scene labeling (Farabet et al., 2013) and shape retrieval (Zhu et al., 2014). Deep learning algorithms, which exploit the unknown structure in the input distribution to discover good representations, have been widely applied in 3D object recognition tasks on 3D data such as 3D models and RGB-D images. Wu et al. (2015) use volumetric Convolutional Neural Network (CNN) architectures on 3D voxel grids to represent a geometric 3D shape for object classification and retrieval. In Zhu et al. (2014), the input is the depth images with different perspectives of 3D objects, and the autoencoder with pre-training by the DBN is applied to extract features. In Xie et al. (2015), an auto-encoder that imposes the Fisher discrimination criterion on the neurons in the hidden layer is used to extract a 3D shape descriptor. In Socher et al. (2012), the convolutional and recursive neural networks are utilized for object reorganization in RGB-D images. There are few studies of point cloud classification using deep learning. Guan et al. (2015) classify 10 species of trees by using DBN for the vertical profile of the tree point clouds. Based on a 2D CNN, a 3D CNN for an object binary classification task with LiDAR data is proposed (Prokhorov, 2010). Maturana and Scherer (Maturana and Scherer, 2015a) introduce 3D CNNs for landing zone detection from LiDAR data. To tackle an object recognition task with LiDAR and RGBD point clouds from different modalities, a volumetric occupancy grid representation is integrated with a supervised 3D CNN to improve the performance in (Maturana and Scherer, 2015b). To make 3D CNN architectures fully exploit the power of 3D representations, Qi et al. (2016) introduce two distinct network architectures of volumetric CNNs for object classification on 3D data. Huang and You (2016) introduce a 3D CNN for recognizing dense voxels generated from point clouds. Qi et al. (2017) further present a deep learning approach, named PointNet, for point cloud classification and segmentation. This approach learns a spatial encoding of each point and then aggregates all the point-based features to a global point cloud feature. It has the ability to directly work on the input point cloud. The main disadvantage is that the PointNet fails to capture local structure induced by the metric space, which constrains its ability to obtain fine-grained patterns to complex scenes. Liu et al. (2017) present a deep reinforcement learning framework for semantic parsing large-scale point clouds. Most of the parameters in the framework are learned, and the class object localization and segmentation are accurate and automatic. A shallow 3D CNN can be well trained on a small 3D voxel grids which are converted from the point cloud. Yet, it is hard to learn discriminative features. In order to improve the training accuracy, we may increase the layers of the 3D CNN, and use of the ResNet (He et al., 2016) to alleviate the vanishing-gradient problem caused by the increase of the CNN layers. However, training the 3D CNN with more layers consumes large computer memories, and often leads to the memory overflow.

3. The DNNSP Framework

In this section, we main describe the process for generating the DNNSP. First, the basic architecture of our approach for point cloud classification is overviewed. Then, the point-based feature representation is derived for representing each point-based feature. Next, the spatial pooling is formed to capture the spatial information of the points. Finally, the setting of the DNNSP is described to implement point cloud classification.

3.1. Basic Architectures for Point Cloud Classification

The features of the points can be directly input to a neural network. However, it is difficult to utilize the spatial structure among the points to achieve high-quality classification results. Thus, we obtain the point cluster-based features to describe the spatial relationships among the points in the point cluster. To achieve this, the point-based feature descriptors of all the points in each point cluster are taken as an input of the DNNSP. The DNNSP then learns the point cluster-based features from the descriptors in each point cluster.

Considered that the terrain points can be separated from the on-ground objects on them (Chen et al., 2013), the aim of this paper is to classify the ground objects on the terrain points. The removal of the terrain points helps to determine the connectivity of on-ground objects. In the on-ground point clouds, we search the k_1 (k_1 is an integer) closest points of each point, and connect the point with its k_1 closest points by edges. In this way, an undirected graph $G(\mathbf{V}, \mathbf{E})$ is generated, where $\mathbf{V}$ is the set of the points and $\mathbf{E}$ is the set of the corresponding edges. The Euclidean distance between two connected points is taken as the weight of the edge. After G is generated, all of the connected components of G can be found. Since objects are often close to others in cluttered urban scenes, a connected component can contain more than one object. In a connected component, a local maximum point may represent the top of an object. To further break the connected component into smaller pieces so that single objects can be isolated, a moving window algorithm is applied to search the local maximum points. When the local maximum points are found, the graph cut (Boykov et al., 2000) is employed to segment the connected component, and the local maximum points are taken as seeds. After the graph cut is performed, the connected component is divided into several point clusters. Each of the point clusters may still contain more than one object. Motivated by the fact that the normalized cut (Shi, J. and Malik, 2000) can aggregate the points with uniform distribution into one cluster, it has been employed here to partition a large point cluster into two new clusters under the condition that the number of points in the cluster is larger than a pre-defined threshold. In this way, we construct multi-level point clusters to capture the coarse to fine parts of the objects. The size of each point cluster is determined by the point density and the object size. Here, three levels are used.

We utilize our previous method (Zhang et al., 2016) to construct the point-based feature descriptors and the point clusters. For each point, a feature vector with 18 dimensions, which contains the eigenvalue feature with 6 dimensions and the spin image feature with 12 dimensions, is computed through its k-nearest neighborhood points, where k = 30, 60 and 90, respectively. Thus, we obtain a 54-dimensional feature vector for each point. Finally, the features of the points in a point cluster are aggregated into an n×54 feature matrix, where n is the point number of the cluster.

The feature matrix is taken as an input of the DNNSP. In the training process, the clusters of all levels are utilized to create a more discriminative feature representation. In the testing stage, we only classify the points in the clusters with the finest level. The aim is to reduce the probability that a cluster contains more than one class.

To address the input, the DNNSP should be invariant to the point permutation and sizes of the input point clusters. Therefore, a simple way to achieve this goal is to determine a point-based feature representation by the multilayer perceptron (MLP) and retain the weight sharing for each point, which means the weight of each point in the same layer is the same. Then, max pooling is employed to aggregate them into the cluster-based features. Finally, the cluster-based features are put into another MLP for point cloud classification. The architectures for the point cloud classification are shown in Fig. 1. In Fig. 1, the point-based feature representation process by the MLP is illustrated in the yellow rectangle. The numbers in the brackets are the layer sizes of the MLP. Based on the architectures, we will expand them in the following sections.

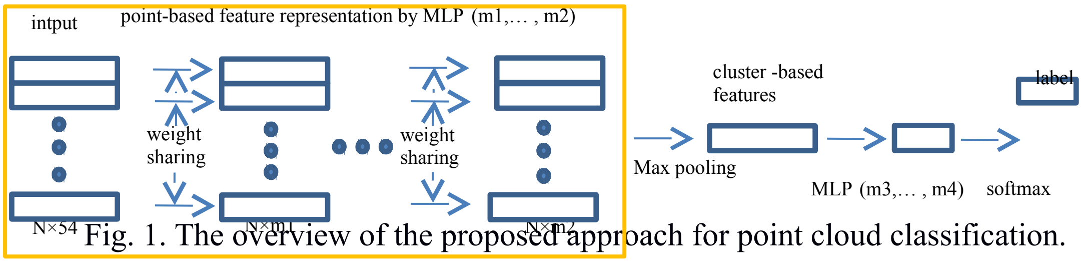


Fig. 1. The overview of the proposed approach for point cloud classification.

3.2. Point-based Feature Representations

The spin images and eigenvalue feature descriptors describe different characteristic of a point. Since the intra-relations between the same type of feature descriptors is closer than the inter-relation between different types of feature descriptors, the representations of the spin images and eigenvalue features are learned separately by the MLP. They are concatenated and then further processed by another MLP to obtain the feature representations of each point in the point clusters. The process is shown in Fig. 2.

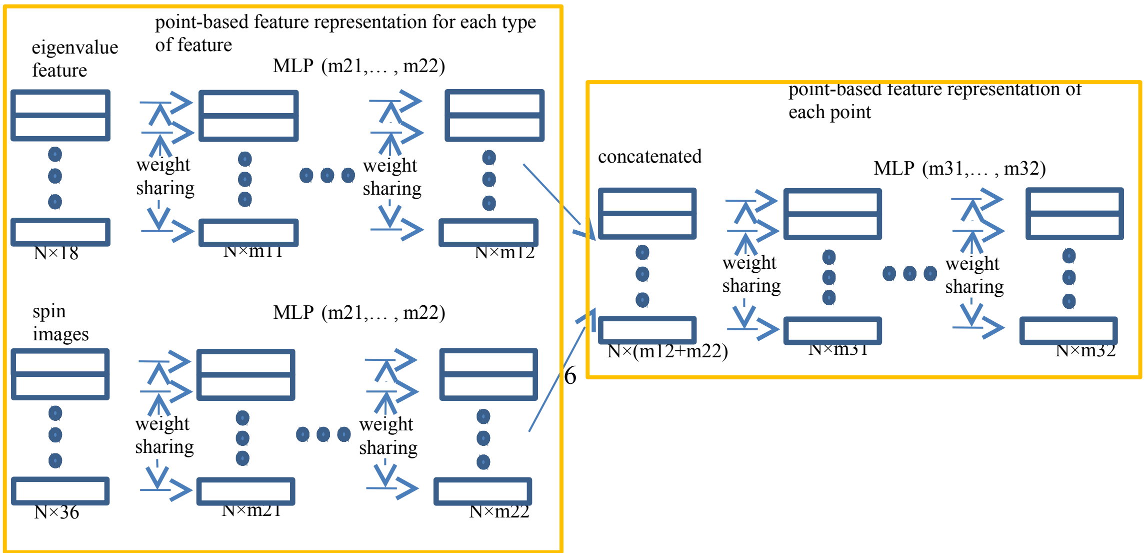


Fig. 2. The point-based feature representation process.

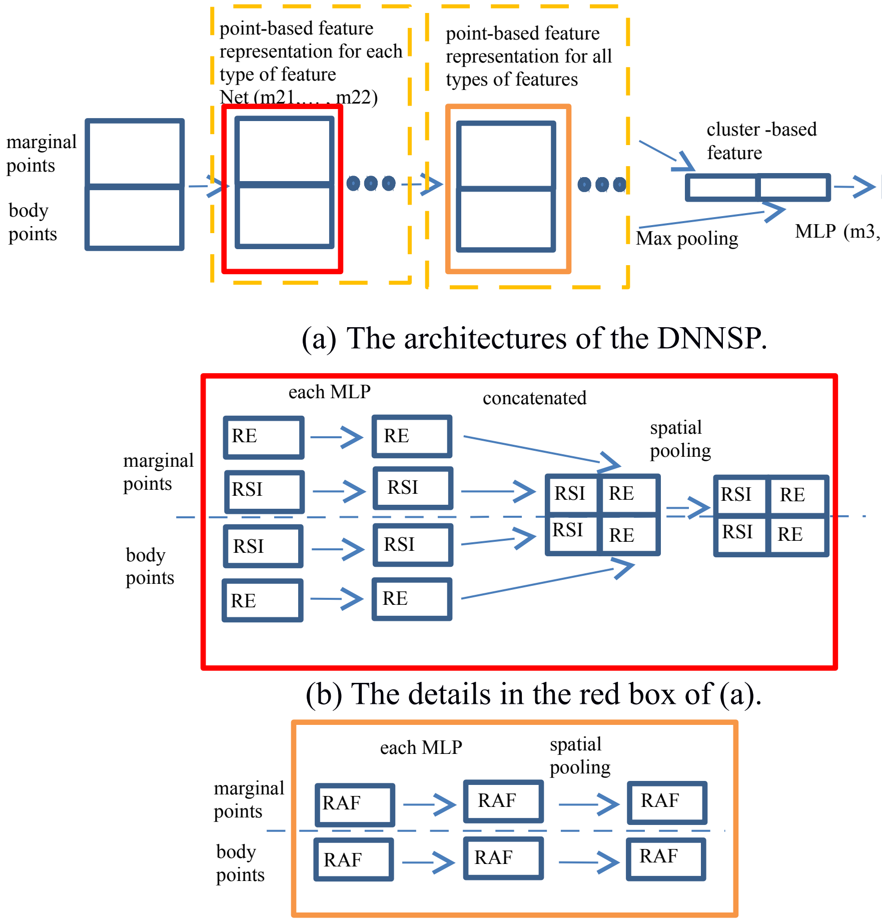
3.3. Spatial Pooling

In the point-based feature descriptors, the spatial relationships among the points have not been considered. It is essential to find the local spatial layout of the points in the point cloud.

To find the spatial layout of the points, the distance minimum spanning tree (DMst) (Wang et al., 2015) is utilized to organize the points in each point cluster by taking the point nearest to the center of the point cluster as the root node. The DMst is a spanning tree, which combines the advantages of the minimum spanning tree (MST) and the spanning tree obtained by the Dijkstra algorithm. The MST is a spanning tree in which the sum of the edge weights is no larger than those of any other spanning tree. Its main advantage is that it can preserve the local spatial structure of the point cloud. The Dijkstra algorithm is a graph-based search algorithm which solves the single-source shortest path problem. It does so by producing a tree that minimizes the sum of the edge weight from each vertex to the single-source vertex, which is the root node. this approach gives a good approximation of tree skeletons even if the point cloud is incomplete or noisy. Unfortunately, the Dijkstra algorithm cannot describe the spatial distribution of points in local regions. Since the DMst integrates the advantages of the above two types of trees, the leaf nodes and the nodes that connect with the leaf nodes in the DMst are usually the marginal points of a point cluster, because the marginal points are usually diffused and far away from the center of the cluster. The remaining nodes are the body points.

Next, we separately handle the body points and marginal points in the DNNSP by configuring different weights for them in the feature representation. In this way, the cluster–based features contain two parts: one part arises from the marginal points and another arises from the body points. A spatial pooling layer is followed behind each layer in the MLP of the point-based feature representation process. In this layer, the average pooling is operated to extract the feature representations of a point (i.e., a node in the DMst) and its connected points. It is noted that the DMst-based spatial pooling is operated on all points in each cluster. The connected points of a DMst node in the cluster can contain the marginal points and body points. Thus, the information of each node within the cluster can flow to the points that are connected to it.

Fig. 3(a) shows the architecture of the DNNSP, and the details in the red and orange boxes are shown in Fig. 3 (b) and (c). To simplify the description, the network in one red box is called Net REF (Net for representation for each type of feature), and the network in one orange box is called Net RAF (Net for representation for each type of feature). In addition, the weights of the same type of features in body points (or marginal points) are shared in each layer of the DNNSP.



(a) The architectures of the DNNSP.

(b) The details in the red box of (a).

(c) The details in the orange box of (a)

Fig. 3 The architecture of the DNNSP. *RE* stands for the eigenvalue feature descriptor. *RSI* stands for the spin image descriptor. *RAF* stands for the final point-based feature representation.

3.4. Implementation

In the DNNSP, the activation function is the *min*(5, elu(*x*)). The method in He et al. (2015) is utilized for initialization, and the Batch Normalization (Ioffe and Szegedy, 2015) is used before the activation. We apply the stochastic gradient descent to train the DNNSP with a mini-batch size of 128. The network learning rate is set to 0.001, and the moment is set to 0.9. In the following experiment, four Net REF and one Net RAF are used in the DNNSP.

4. Experimental Results

To evaluate the performance of the DNNSP, the DNNSP is employed to classify point clouds of six urban scenes. We also compare the DNNSP with the following four methods.

The first method (Method I) is the one described in Wang et al. (2015). This method employs a multi-scale and hierarchical framework to classify point clouds of cluttered urban scenes. In this framework, the features of point clusters are constructed by employing the Latent Dirichlet Allocation (LDA).

The second method (Method II) is the one described in Zhang et al. (2016). In this method, the point cloud is split into hierarchical clusters of different sizes. Then, LDA and sparse coding are jointly performed to extract and encode the shape features of the multilevel point clusters. The features at different levels are used to capture information on the shapes of objects of different sizes.

The third method (Method III) is the one described in Guo et al. (2015). In this method, each point is associated with a set of derived features using geometric, multi-return and intensity information, and the features are selected using JointBoost to evaluate their correlations.

The fourth method (Method IV) is the one described in Li et al. (2016). In this method, a set of point-based descriptors for recognizing urban point clouds is constructed. The initial 3D labeling of the categories is generated by utilizing a linear SVM classifier on the descriptors. These initial classification results are globally optimized by the Multi-label Graph-cut approach, and then are further refined automatically by a local optimization approach based upon the object-oriented decision tree.

4.1. Experimental Datasets

The point clouds of six urban scenes are used in the experiment.

Scenes I and II: The two scenes come from Tianjin city, China. The point clouds contain buildings, trees and a few cars. The point density is 20–30 points/m^2. The eaves extend outside the building roofs. Due to scattering, numerous noisy points occur around the eaves, which causes the eaves to be easily misclassified.

Scene III: The point cloud is the Vaihingen dataset (Niemeyer et al., 2014) provided by International Society for Photogrammetry and Remote Sensing, which are covered by 10 strips. The average strip overlap is 30%. The point cloud mainly contains four categories, i.e. roofs, facades, shrubs and trees, and low vegetation. The point density varies considerably over the entire block depending on the overlap. In the regions covered by only one strip, the mean point density is 4 point/m^2.

The datasets of Scenes I–III are the airborne laser scanning (ALS) point clouds, obtained by a Leica ALS50 system with a mean flying height of 500 m above the ground and a 45° field of view.

Scenes IV–VI: The datasets are the terrestrial laser scanning (TLS) point clouds provided by Eidgenössische Technische Hochschule Zurich (Hackel et al., 2017). The point density is uneven. The point clouds contain natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artefacts and cars.

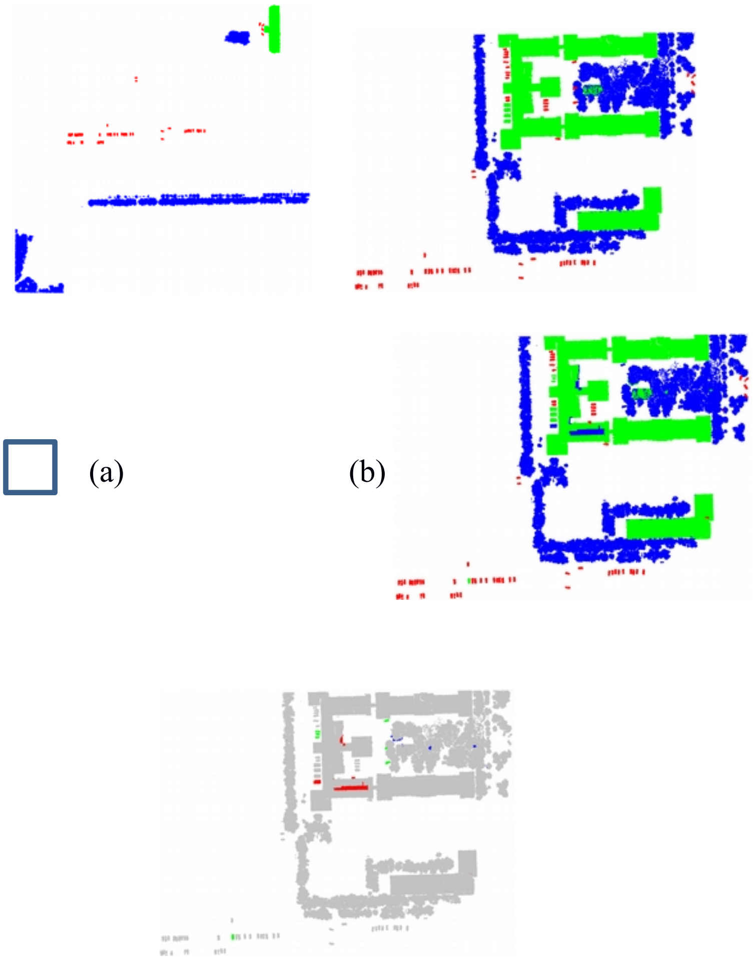
4.2. Classification Results on the ALS Point Clouds

For Scenes I and II, the method in (Zhang et al., 2016) is utilized to generate the feature representation and clustering results. For Scene III, we divide the scene into two separate parts: one for training and the other for testing. The point cloud is clustered into multi-level point clusters whose sizes are 60, 120 and 240.

Table 1 lists the number of points in the training and testing data of Scenes I–III. Table 2 lists the classification accuracy of the three scenes. Figs. 4–6 illustrate the training data, testing data and classification results of the three scenes. In Figs. 4 and 5, the green points are on the buildings; the blue points are on the trees, and the red points are on the cars. In Fig. 6, the navy blue points are on the roofs; the orange points are on the facades; the light blue points are on the low vegetation; and the green points are on the shrubs and trees. In Figs. 4–6(d), the gray points are the correctly classified points.

Table 1. The number of points in Scenes I-III.

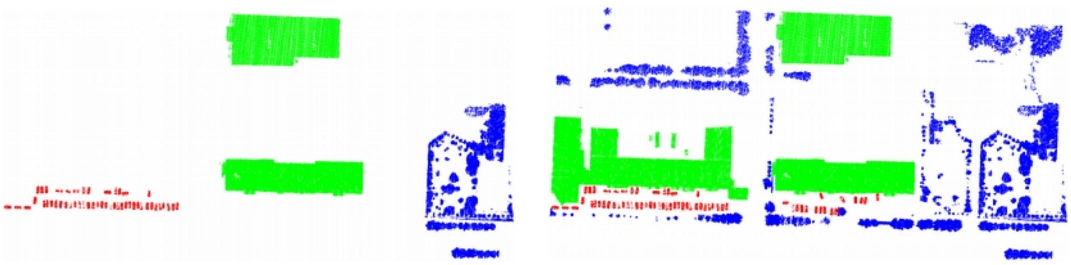
<table>
<tr><td>Scene I</td><td colspan="2">Building</td><td colspan="2">Tree</td><td colspan="2">Car</td></tr>
<tr><td>Training data</td><td colspan="2">37847</td><td colspan="2">70540</td><td colspan="2">5410</td></tr>
<tr><td>Testing data</td><td colspan="2">201674</td><td colspan="2">218110</td><td colspan="2">7987</td></tr>
<tr><td>Scene II</td><td colspan="2">Building</td><td colspan="2">Tree</td><td colspan="2">Car</td></tr>
<tr><td>Training data</td><td colspan="2">64952</td><td colspan="2">39743</td><td colspan="2">4584</td></tr>
<tr><td>Testing data</td><td colspan="2">157447</td><td colspan="2">74264</td><td colspan="2">7738</td></tr>
<tr><td>Scene III</td><td>Roof</td><td colspan="2">Shrub and tree</td><td colspan="2">Low vegetation</td><td>Façade</td></tr>
<tr><td>Training data</td><td>72582</td><td colspan="2">120009</td><td colspan="2">113344</td><td>19758</td></tr>
<tr><td>Testing data</td><td>79463</td><td colspan="2">62769</td><td colspan="2">67506</td><td>7492</td></tr>
</table>



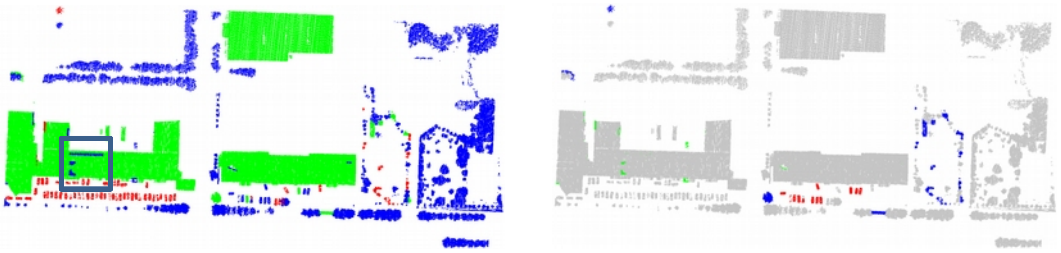
(a) (b)

(c) (d)

Fig. 4. The training data, testing data and classification results for Scene I. (a) Training data. (b) Testing data. (c) Classification results. (d) Highlighted incorrectly classified points. The green points are on the buildings. The blue points are on the trees. The red points are on the cars. The gray points are the correctly classified points in (d).

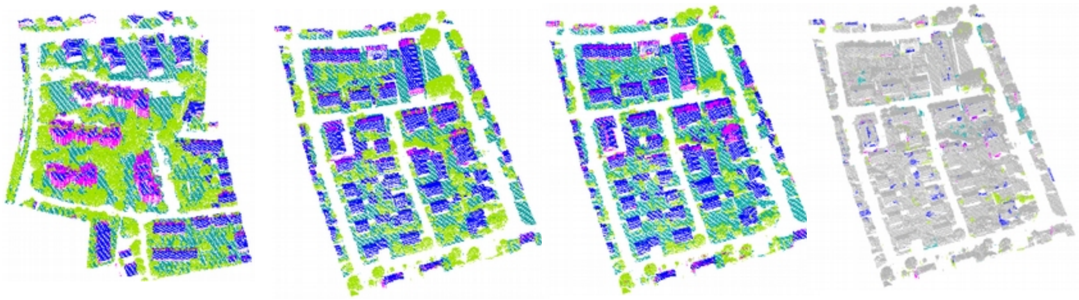


(a) (b)



(c) (d)

Fig. 5. The training data, testing data and classification results of Scene II. (a) Training data. (b) Testing data. (c) Classification results. (d) Highlighted incorrectly classified points. The green points are on the buildings. The blue points are on the trees. The red points are on the cars. The gray points are the correctly classified points in (d).



(a) (b) (c) (d)

Fig. 6. The training data, testing data and classification results of Scene III.. (a) Training data. (b) Testing data. (c) Classification results. (d) Highlighted incorrectly classified points. The navy blue points are on the roofs. The orange points are on the facades. The light blue points are on the low vegetation. The green points are on the shrubs and trees.

Table 2. Comparisons of the classification results of Scenes I-III in terms of precision/recall and accuracy.

Scene I	Building(%)	Tree(%)	Car(%)	Accuracy(%)
our Method	**97.7/98.8**	**99.2/97.7**	**85.2/98.1**	**98.2**
Method I	94.0/95.4	95.0/94.3	79.1/60.8	94.5
Method II	95.7/96.2	95.9/95.9	80.8/67.9	95.8
Method III	89.7/98.1	97.9/89.1	65.2/46.6	92.9
Method IV	93.5/96.2	95.3/94.1	75.3/84.6	95.1
Scene II	Building(%)	Tree(%)	Car(%)	Accuracy(%)
our Method	**98.9/98.4**	96.2/96.5	**78.4/84.9**	**97.4**
Method I	90.3/93.9	97.6/96.5	49.4/42.0	94.1
Method II	94.7/94.5	**98.1/97.7**	53.9/60.5	95.5
Method III	86.8/91.2	96.8/95.5	44.1/34.8	92.2

Method IV	92.7/94.0	95.1/92.6	71.2/65.3	94.3

Scene III	Roof (%)	Shrubs and trees (%)	Low vegetation (%)	Facade (%)	Accuracy (%)
our Method	**91.1/96.5**	**92.5/88.9**	**94.3/93.7**	**74.1**/61.8	**91.9**
Method I	75.9/94.1	89.4/72.9	89.1/84.7	51.6/75.7	83.1
Method II	79.7/97.4	86.2/70.0	92.0/87.0	42.3/**90.3**	84.1
Method III	73.4/91.3	87.8/71.6	87.6/82.3	24.9/51.8	80.3
Method IV	77.8/92.9	86.9/83.8	89.5/74.2	43.6/65.9	82.8

Method I: the method in Wang et al. (2015). Method II: the method in Zhang et al. (2016) . Method III: the method in Guo et al. (2015). Method IV: the method in Li et al. (2016).

As shown in Figs. 4–6, most of the points are classified correctly, indicating that the DNNSP can extract good cluster features for the classification. In Scenes I and II, only the blue blocks in Figs. 4(c) and 3(c) are misclassified. In the blue block of Fig. 4(c), because of the large amount of noise around the eaves, these points look like they are on a crown. In the blue block of Fig. 5(c), there is a line structure isolated from the building. The line structure may be on an edge of an eave, and most of the points on it are misclassified. The classification accuracy of the cars is lower than those of the buildings and trees. The reason is that there are not sufficient car points in the training data. This causes the car features of the DNNSP are not well trained. In Fig. 6, it is observed that most of the misclassified points occur at the object borders, such as the roofs crowding with the trees or borders between the roofs and facades. In these cases, the neighboring points may be on objects of a different class, which causes the point-based features themselves to be less discriminative. However, except for the border points between the roofs and facades, most of the facade points have been recognized correctly.

Because our method can learn more robust feature representations than other methods, it achieves the highest classification accuracy except for the tree category in Scene II and the recall of the facades in Scene III. In the three scenes, it is noted that the classification accuracies of the cars and facades achieved using our method are also higher than those by using other methods, indicating that our method is competitive for classifying the categories with a few points. In Scene III, many of the roof and facade points are misclassified using other methods. The reason is mainly because the roof points are confused with the low vegetation points and the facade points are confused with the shrubs and trees points. However, the four categories are all classified better than other methods, which verifies that our method can distinguish the categories even though they look similar in shapes.

4.3. Classification Results of the TLS Point Clouds

For Scenes IV–VI, the point clouds are divided into multi-level clusters with the sizes of 100, 300 and 500. To more clearly show the generalization ability of our method, one-third of the clusters that belong to the size of 500 are taken as the training data, and the point clouds in Scenes V and VI are used for the testing data. Table 3 lists the number of points in Scenes IV–VI. Table 4 lists the classification accuracy of Scenes V and VI. In Fig. 7, the light green points are on a natural terrain; the dark green points are on high vegetation; the bright green points are on low vegetation; the red points are on buildings; the purple points are on hard scape; the orange points are on the scanning artefacts; and the pink points are on cars.

Table 3. The number of the points in Scenes IV-VI.

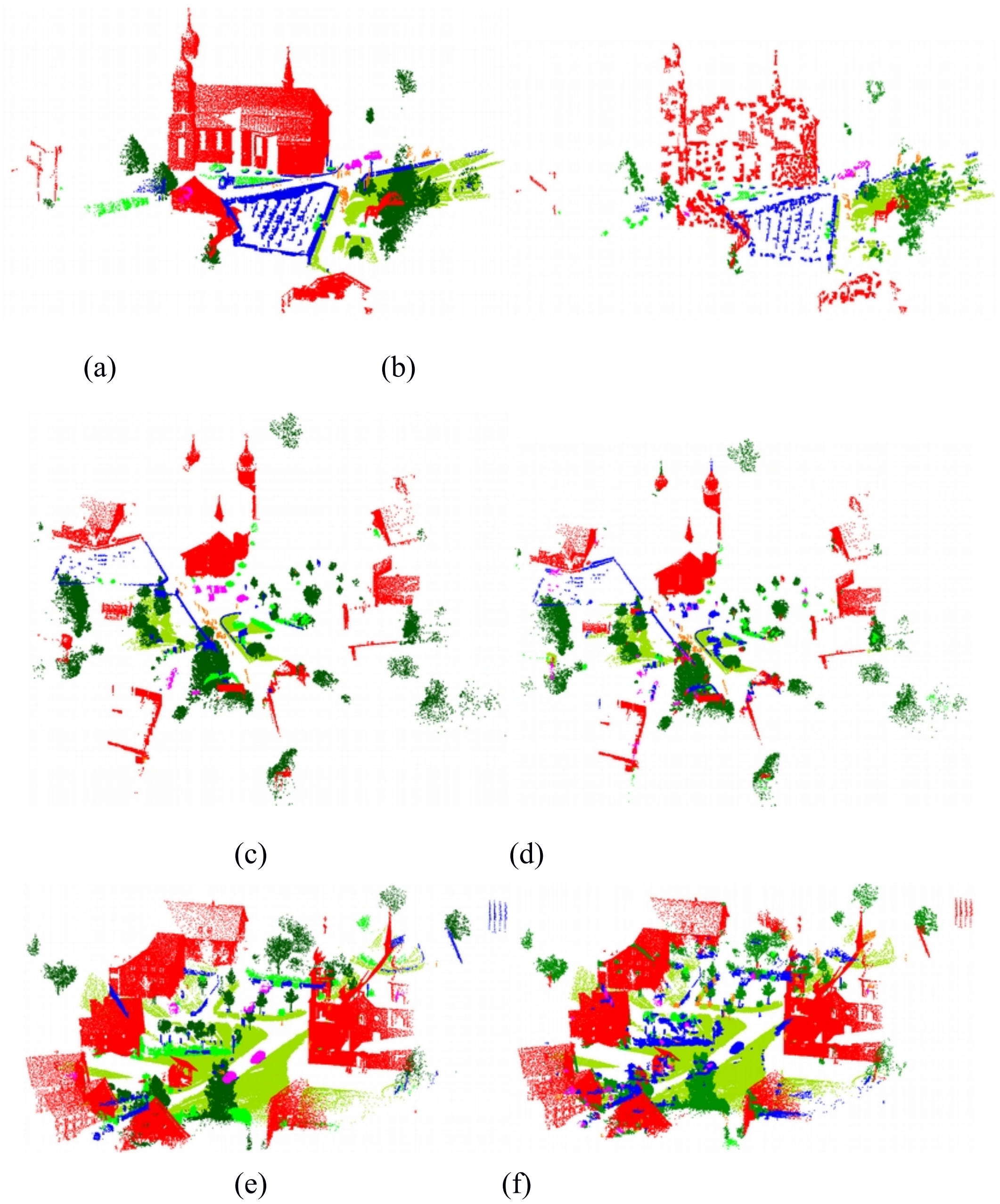
	Natural terrain	High vegetation	Low vegetation	Building	Hard scape	Scanning artefacts	Car
Scene IV	3174149	1027837	592309	539935	1260888	7040	92873
Scene V	3507576	2537763	49680	1241838	762982	13899	65636
Scene VI	4924691	352455	172081	1611908	46140	751	403970

Table 4. Comparisons of the classification results of Scenes V-VI in terms of Precision/recall and accuracy.

Scene V	Natural terrain	High vegetation	Low vegetation	Building	Hard scape	Scanning artifact	Car	Accuracy
our Method	**99.3**/ **97.4**	**94.7**/ **96.0**	**53.3**/ **89.3**	**93.9**/ **84.8**	**89.9**/ **78.5**	**88.2**/ **73.5**	11.6/ 26.1	**91.1**
Method I	97.7/ 76.5	83.6/ 95.2	15.8/ 48.5	72.4/ 71.5	82.9/ 71.5	37.0/ 12.9	27.1/ 68.0	84.0
Method II	96.7/ 86.6	83.8/ 95.2	45.8/ 82.3	72.8/ 73.6	82.1/ 78.2	44.2/ 17.9	33.7/ 49.7	84.5
Method III	96.1/ 86.4	79.7/ 95.0	45.9/ **89.3**	70.4/ 69.4	80.1/ 75.8	10.5/ 2.8	**34.9/ 75.4**	83.6
Method IV	99.2/ 98.8	89.4/ 85.7	6/ 56.7	88.9/ 74.1	88.1/ 66.4	52.7/ 12.9	13.5/ 54.2	85.3
Scene VI	Natural terrain	High vegetation	Low vegetation	Building	Hard scape	Scanning artifact	Car	Accuracy
our Method	**98.6**/ **99.5**	**88.1**/ **66.4**	4.0/ 31.1	**92.7**/ **98.4**	**66.6**/ 5.2	43.8/ 2.1	1.3/ 35.1	**89.3**

Method I	76.9/ 94.7	72.1/ 17.7	14.8/ 32.5	80.5/ 79.8	28.7/ 7.4	16.3/ 3.8	1.4/ 14.6	71.4
Method II	75.9/ 92.5	78.8/ 16.3	**26.5/ 46.1**	70.1/ 81.9	39.6/ **9.6**	30.5/ 5.6	3.6/ 31.4	69
Method III	78.2/ 93.2	82.3/ 14.7	19.1/ 40.2	65.8/ 82.4	32.4/ 8.7	**59.2/ 10.7**	**5.2/ 39.7**	69.3
Method IV	84.6/ 93.7	85.4/ 19.0	16.4/ 37.5	69.3/ 83.5	30.8/ 7.8	44.9/ 8.0	4.3/ 35.5	74.2

Method I: the method in Wang et al. (2015). Method II: the method in Zhang et al. (2016) . Method III: the method in Guo et al. (2015). Method IV: the method in Li et al. (2016).



(a) (b)

(c) (d)

(e) (f)

Fig. 7. A portion of the point clouds of Scenes IV-VI, training data and classification

results of Scenes V and VI. (a) Point cloud in Scene IV. (b) Training data. (c) Point cloud in Scene V. (d) Classification results of (c). (e) Point cloud in Scene VI. (f) Classification results of (e). The light green points are on the natural terrain. The dark green points are on high vegetation. The bright green points are on low vegetation. The red points are on buildings. The purple points are on hard scape. The orange points are on the scanning artefacts. The pink points are on cars.

The classification results on Scenes V and VI are not as good as those of Scenes I–IV. In the two scenes, the shape features of some of the categories are easily confused, such as high vegetation and low vegetation, hard scape and the other categories, especially for the hard scape, which is a class that is not taken as a special object. The hard scape contains rocks that mingle with cars, fences that mingle with vegetation, steles that mingle with cars or buildings, and so on. Even worse, there are many hard scape points in Scene IV. To fit these points, the DNNSP is overfitting. All of the above reasons lead to the low classification accuracy for the hard scape. Most of the car and low vegetation points are classified into the hard scape. Although there are many car points, only three car samples are in Scene I. The training data of the cars is not sufficient. Additionally, the cars are similar to rocks. Therefore, the performance on car classification is not good. The vegetation fences belong to low vegetation, but the fences made of other materials belong to hard scape. They are very similar in the point clouds; thus, the classification performance of the class fences is also not good. The high vegetation and low vegetation are also confused. The reason is that there are no clearly defined differences between them in terms of the shape. If the height is used, then performance will be improved. The natural terrain and buildings are classified correctly, which indicates that the accuracy is high if the sample size is sufficiently large.

Compared with the other methods, our method achieves the highest classification accuracy. Most notably, the accuracy improves at least 20% for Scene VI. This finding means that the DNNSP has the ability to learn better feature representations from the point-based features and improve the classification accuracy. Moreover, compared to the improvement for ALS point clouds by the DNNSP, the improvement is more obvious in complicated scenes.

4.4. Performance of the architectures in the DNNSP

4.4.1 Classification performances of Net REF, Net RAF and spatial pooling

Different numbers of Net REF and Net RAF are utilized to present the influences of the nets on the classification.

Table 5 lists the classification accuracy (%) of Scenes I-VI through the DNNSP with 0–4 Net REF and 0–3 Net RAF, with/without spatial pooling. The lowest classification results are expressed using the actual accuracy, and are highlighted by underlines. Other classification results are shown by the relative improvements to the lowest classification accuracy. The highest classification results are highlighted in bold. The poor classification results are replaced by "-". As listed in Table 5, for the classification results obtained using the DNNSP with/without spatial pooling, Net REF is helpful, and the best results are all obtained by using at least one Net REF. This finding means that the use of intra-relations of the point-based features in the Net REF is helpful for improving the classification performance. One Net RAF is sufficient in the classification. In most cases, the more Net RAFs are applied, the worse the classification results become. When the number of points increases in Scenes III, V and VI, the Net RAF improves the classification performance. This indicates that the intra-relations of the point-based features still have a little positive influence on the classification. Additionally, simply concatenating the two types of features, i.e. spin images and eigenvalue features, into a vector is not a good idea. In future work, we will find a better way to mine the inter-relations. In Scenes I-III, sometimes the classification accuracy has a sudden decrease as the number of nets changes. This situation does not occur in Scenes V and VI. We argue that the points in Scenes I-III are few and the DNNSP can easily converge to a local minimum or overfitting.

Table 5. The classification accuracy (%) through the DNNSP with/without spatial pooling.

Scene I					
Net REF Net RAF	Without Net REF	One Net REF	Two Net REFs	Three Net REFs	Four Net REFs
Without Net RAF		+2.9 /+3.7	+3.1 /+3.9	+3.1 /+4.2	+3.1 /**+4.4**
One Net RAF	93.8 /-	-/-	+3.8/+0.2	.+3.2 /+2.1	+3.2 /-
Two Net RAFs	**+3.9**/+1.6	+2.9/-	+2.9 /-	+3.5 /+1.4	+2.9 /+3.3
Three Net RAFs	+0.4/+1.5	-/+2.4	+0.9 /+2.6	+0.3/+3.5	-/+2.6
Scene II					
Without Net RAF		+2.4 /+4.7	+0.9 /+4.7	+0.5 /+5.2	+0.3 /+**5.4**
One Net RAF	- /-	**+3.9**/-	+1.9 /+2	+2.3 /+4	+1.9/-
Two Net RAFs	- /+3.4	- /-	+3.1/-	92.6 /+4.7	+2.6 /+3.4
Three Net RAFs	-+0.4	- /+2.1	- /+3.4	- /+2.2	+1 /-
Scene III					

Without Net RAF		+0.8/+0.5	+1.6/+1.5	**+2**/+1.8	+1.4/+**2.4**
One Net RAF	+0.5/88.9	+1.7/+1	+1.4/+1.5	**+2**/+1.6	+1.6/+1.8
Two Net RAFs	+1.8/+1.4	+1.7/+1.4	+1.9/+1.5	+1.8/+1.8	+1.9/+1.7
Three Net RAFs	+1.8/+1.3	+1.7/+1.6	+1.5/+1.7	+1.6/+1.8	+1.9/+1.2
Scene V					
Without Net RAF		+4.8 /+4.8	- /-	+7.7 /+2.8	+7.1/+6.9
One Net RAF	+4.5 /+4.3	+4.7 /+6.1	+8.9/+8.8	+8.2/+9.3	**+10.2**/+**11**
Two Net RAFs	80.8 /+0.8	+5.7+3.6	+2.9 /+4.9	+6.8/+7.3	+6.8/+8.5
Three Net RAFs	+2 /+1.5	+1.4/+3	+6.4 /+3.9	+8.3 /+6	+5.7/+6.7
Scene VI					
Without Net RAF		+6.2/+6.1	+6.1 /+6.1	+6.8 /+6.8	+6.6/+7.4
One Net RAF	+4.1 /+4.4	+6.2/+7.1	**+7.7** /+7.2	+7.1 /+7.4	+5.5/+7.7
Two Net RAFs	+1.6 /+3.7	+4.3/+**8.8**	+7.4/+6.7	+6.3 /+6.2	+6.1 /+6.6
Three Net RAFs	+4.4 /+4.9	80.4/+4.5	+2.6 /+6.8	+4.6 /+7	+7.2/+6.8

The DNNSP with spatial pooling achieves better classification results than the DNNSP without spatial pooling, and it also enhances the classification performance of all of the scenes. When the number of Net RAFs is determined, the classification accuracy with spatial pooling is enhanced with an increase in the number of Net REFs, but the classification accuracy without spatial pooling is random. It is concluded that with the help of spatial pooling, the DNNSP can be stacked deeper to obtain better classification performance through more Net REFs.

4.4.2 Classification performance by separating the body and margin points

To show the advantages of separately using the body and margin points for the point cloud classification, we use all of the points without distinguishing the body points and margin points to classify the scenes. We observe from Table 6 that the classification performance obtained using the body and margin point separately is improved for all of the scenes, especially for Scenes V and VI which are more complex.

Table 6. The classification accuracy obtained by using all of the points without distinguishing the body points and margin points.

Scene I	Scene II	Scene III	Scene V	Scene VI
98.0%	97.5%	89.6%	86.3%	82.9%

4.4.3 Classification performance by using all of the levels of clusters

To evaluate the effectiveness of the extracted common features in the DNNSP, we take the point clusters with the smallest size as the input. The classification results are listed in Table 7. From Table 7, it is observed that the classification accuracies are obviously lower than those obtained using all levels in Scenes I–III, which shows that the common features are helpful for the classification.

Table 7. The classification accuracy obtained by using clusters with one level.

Scene I	Scene II	Scene III	Scene V	Scene VI
95.6%	94.7%	89.2%	90.3%	88.4%

5. Conclusions

The features learned by most of the current deep learning methods can obtain high-quality image classification results. However, these methods are hard to be applied to recognize 3D point clouds due to unorganized distribution and various point density of data. In this paper, we have presented the DNNSP to classify outdoor point clouds without rasterization. To ensure that the point-based representations are discriminative and robust, the DMst-based pooling utilizes spatial information among points in the point clouds. The body points and marginal points in the DNNSP are handled separately by configuring different weights for them in the feature representation. In this way, the DNNSP can learn the feature representations of points from multiple levels, which makes the point cluster-based feature representations robust and discriminative. The experimental results demonstrate the effectiveness of the DNNSP for point cloud classifications.

In future work, we will extend the DNNSP to directly learn features from the raw point clouds, and employ our method to other applications such as 3D object recognition or retrieval.

References

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, pp. 1798-1828.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), pp.1222-1239.

Brodu, N., Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. ISPRS Journal of Photogrammetry and Remote Sensing, 68, pp. 121-134.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38, W8.

Chen, D., Zhang, L., Wang, Z., Deng, H., 2013. A mathematical morphologybased multi-level filter of LiDAR data for generating DTMs. Science China Information Sciences, 56(10), pp. 1–14.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, pp. 1915-1929.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors, In: European Conference on Computer Vision, pp. 224-237.

Fukano, K., Masuda, H., 2015. Detection and Classification of Pole-Like Objects from Mobile Mapping Data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2, pp. 57-64.

Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments, In: IEEE International Conference on Computer Vision, pp. 2154-2161.

Guan, H., Yu, Y., Ji, Z., Li, J., Zhang, Q., 2015. Deep learning-based tree classification using mobile LiDAR data. Remote Sensing Letters, 6, pp. 864-873.

Guo, B., Huang, X., Zhang, F., Sohn, G., 2015. Classification of airborne laser scanning data using JointBoost. ISPRS Journal of Photogrammetry and Remote Sensing, 100, pp. 71-83.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. SEMANTIC3D NET: A new large-scale point cloud classification benchmark, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1-W1, pp.91-98.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, In: IEEE International Conference on Computer Vision, pp. 1026-1034.

He, K., Zhang, X., Ren, S., Sun J., 2016. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Huang, J., You, S., 2016. Point cloud labeling using 3D convolutional neural network, Proc. of the International Conf. on Pattern Recognition.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21, pp. 433-449.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, In: Advances in Neural Information Processing Systems, pp. 1097-1105.

Kragh, M., Jørgensen, R.N., Pedersen, H., 2015. Object detection and terrain classification in agricultural fields using 3D lidar data, In: International Conference on Computer Vision Systems, pp. 188-197.

Koppula, H.S., Anand, A., Joachims, T., Saxena, A., 2011. Semantic labeling of 3d point clouds for indoor scenes, In: Advances in Neural Information Processing Systems, pp. 244-252.

Li, Z., Zhang, L., Tong, X., Du, B., Wang, Y., Zhang, L., Zhang, Z., Liu, H., Mei, J., Xing, X., 2016. A Three-Step Approach for TLS Point Cloud Classification. IEEE Transactions on Geoscience and Remote Sensing, 54, pp. 5412-5424.

Liu, F., Li, S., Zhang, L., Zhou, C., Ye, R., Wang, Y., Lu, J., 2017. 3DCNN-DQN-RNN: A Deep Reinforcement Learning Framework for Semantic Parsing of Large-scale 3D Point Clouds. IEEE International Conference on Computer Vision.

Maturana, D., Scherer, S., 2015a. 3d convolutional neural networks for landing zone detection from lidar, In: IEEE International Conference on Robotics and Automation, pp. 3471-3478.

Maturana, D., Scherer, S., 2015b. Voxnet: A 3d convolutional neural network for real-time object recognition, In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 922-928.

Niemeyer, J., Rottensteiner, F., Sörgel, U., 2014. Contextual classification of lidar data and building object detection in urban areas, ISPRS Journal of Photogrammetry and Remote Sensing, 87, pp. 152-165.

Niemeyer, J., Rottensteiner, F., Soergel, U., Heipke, C., 2016. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Congress, pp. 12–19.

Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing, 66, pp. S28-S39.

Prokhorov, D., 2010. A convolutional learning system for object classification in 3-D LIDAR data. IEEE Transactions on Neural Networks, 21, pp. 858-863.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J., 2016. Volumetric and multi-view cnns for object classification on 3d data, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5648-5656.

Shi, J. and Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888-905.

Socher, R., Huval, B., Bath, B.P., Manning, C.D., Ng, A.Y., 2012. Convolutional-Recursive Deep Learning for 3D Object Classification. In: Advances in Neural Information Processing Systems, p. 8.

Stuhlsatz, A., Lippel, J., Zielke, T., 2012. Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Transactions on Neural Networks and Learning Systems, 23, pp. 596-608.

Wang, Z., Zhang, L., Fang, T., Mathiopoulos, P.T., Tong, X., Qu, H., Xiao, Z., Li, F., Chen, D., 2015. A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. IEEE Transactions on Geoscience and Remote Sensing, 53, pp. 2409-2425.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Computers & Graphics, 49, pp. 47-57.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes, In: the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912-1920.

Xie, J., Fang, Y., Zhu, F., Wong, E., 2015. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval, In: the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1275-1283.

Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M., 2011. 3-d scene analysis via sequenced predictions over points and regions. In: IEEE International Conference on Robotics and Automation, pp. 2609-2616.

Xu, S., Wang, R., and Zheng, H., 2017. Road Curb Extraction from Mobile LiDAR Point Clouds. IEEE Transaction on Geoscience and Remote Sensing, 55(2), pp. 996–1009.

Yang, B., Dong, Z., Zhao, G., and Wenxia Dai, W., 2015a. Hierarchical Extraction of Urban Objects from Mobile Laser Scanning Data, ISPRS Journal of Photogrammetry and Remote Sensing, 99, pp. 45-57.

Yang, B., Zang, Y., Dong, Z., and Huang, R.,2015b. An Automated Method to Register Airborne and Terrestrial Laser Scanning Point Clouds, ISPRS Journal of Photogrammetry and Remote Sensing, 109, pp. 62-76.

Zhang, J., Lin, X., Ning, X., 2013. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. Remote Sensing, 5, pp. 3749-3775.

Zhang, R., Candra, S.A., Vetter, K., Zakhor, A., 2015. Sensor fusion for semantic segmentation of urban scenes. In: IEEE International Conference on Robotics and Automation, pp. 1850-1857.

Zhang, Z., Zhang, L., Tong, X., Mathiopoulos, P.T., Guo, B., Huang, X., Wang, Z., Wang, Y., 2016. A multilevel point-cluster-based discriminative feature for ALS point cloud classification. IEEE Transactions on Geoscience and Remote Sensing, 54, pp. 3309-3321.

Zheng, H., Wang, R., and Xu, S., 2017. Recognizing Street Lighting Poles from Mobile LiDAR Data, IEEE Transaction on Geoscience and Remote Sensing, 55(1), pp. 407-420.

Zhu, Z., Wang, X., Bai, S., Yao, C., Bai, X., 2014. Deep learning representation using autoencoder for 3D shape retrieval, In: International Conference on Security, Pattern Analysis, and Cybernetics, pp. 279-284.