A Deep Neural Network with Spatial Pooling for Outdoor Point Cloud Classification

Anonymous CVPR submission

Paper ID 1246

Abstract

Large-scale urban scenes usually contain a large number of object categories and many overlapped or closely neighboring objects. All of these pose great challenges in point cloud classification. To extract discriminative features, the deep learning has been employed, but conventionally, point clouds are rasterized first to construct spatial structure for point clouds. Yet, rasterization is hard to be designed for all the objects in the large-scale outdoor point cloud and a lot of the shape and geometric information is lost in the rasterization process. To address problem, a deep neural network with spatial pooling (DNNSP), which can directly classify large-scale point clouds and learn the spatial relationships among points, is proposed in this paper. The strategy that represents the point features and then pools them on average to cluster features makes sure the DNNSP can directly handle the point clouds. 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. It can learn the representations of points scaled from the whole region to the center of the point clusters, which makes the representations robust and discriminative. Our method achieves high classification performance on different types of point clouds and significantly outperforms other methods

1. Introduction

Due to the increasing availability of 3D point cloud data and respective acquisition systems, the ability to classify 3D point clouds of large-scale urban scenes efficiently and accurately is of major importance in remote sensing and computer vision fields. Much research work has focused on this problem in the past decade. Most of the developed approaches [5, 16, 26, 30] usually first extract and aggregate hand-designed features, and fed them into an off-the-shelf classifier. Although the hierarchy features [25, 29, 31] extracted from hand-designed features are also used, the features are not as powerful as the very deep feature representations from the deep learning [1, 6, 10, 13]. With the help of rasterization, spatial relationships are constructed for point clouds and the deep learning technology is fit to the rasterized point clouds or 3D models [17, 18, 27, 32]. However, the methods focused on the indoor applications for 3D model recognition [18, 27, 32] or detected landing zones but not classification different objects from LiDAR data [17]. Different rasterizations are carefully designed for their target objects. Such methods can work well for dense and even point clouds, but limits for large-scale urban scenes in which the rasterization is hard to be designed for all the objects with uneven point density and missing data. The rasterization process will also lose a lot of valuable information about the shape and geometric layout of objects. With the original 3D point cloud data, we can more precisely determine the shape, size and geometric orientation of the objects [15]. Moreover, augmenting spatial cues with 3D information can enhance object detection in cluttered, real world environments [8]. Consequently, there still remains obvious gap between deep learning and large-scale point clouds. Its potential for point cloud classification is still relatively unexplored.

The main difficulty for classifying points by the deep learning stems from the unorganized point clouds. Different from images whose relationships among pixels can be caught by the sliding windows, the points in a point clouds have no clearly neighborhoods. Although the nearest k points can be seen as an approximation, there is no explicit rank for the k points. Moreover, the nearest k points are affected by incomplete data and uneven point density which depends on the distance to the sensor and the surface orientation. As point clouds only supply the positions of points, we cannot classify a point by its position but by the shape of the points around it. Therefore, we cluster the points to point clusters and describe the point cluster features through the spatial relationships among points in them. The advantage of the clusters is the flexibility, compared to the rasterization, especially for the uneven point density. A cluster

can contain various sizes of regions, but the pixel size of rasterization is fixed.

In this paper, a deep neural network with spatial pooling (DNNSP) that exploits rich relational information of the points is proposed to classify a raw 3D point cloud without rasterization. To avoid the rasterization, we learn the point-based feature representations and then perform average pooling for the representations of clusters to learn the cluster-based features. To utilize the spatial structures among the points, we present the distance minimum spanning tree (DMst) [25] based pooling.

At preprocessing, the raw point cloud is clustered into point clusters, and the feature of each point is computed. They are taken as the input of the DNNSP. The weight sharing strategy for points is applied in point-based feature representation process. The point-based representations in the clusters are pooled on average to cluster-based features. In this way, a simple framework which can be used directly for point clouds classification without rasterization is constructed. However, without the use of the spatial structure, the most important information of point clouds is not used. Also it will be validated in experiments the simple framework cannot become a good deep neural network framework, because the accuracy changes randomly by the increasing of the depth.

Compared to the pooling windows in the image classification, the connected points are similar as the pixels in the windows, but how many and which points chosen are not explicit. The DMst can describe the local spatial structure, so a point and its connection points are seen as the window. The DMst-based pooling also inherits this advantage, which makes the representations robust. Similar as the effect of Pooling in image, DMst-based pooling helps the representations be learnt scaled from the whole region to the center of the cluster. The DMst can also separate the points into marginal points and body points. The body points are helpful to classify the objects with non-resemblance in appearance, and the marginal points are helpful to classify similar objects. The marginal points and body points have different weights in the DNNSP, but their weights can be propagated to each other in the pooling process. In this way, the contributions of the marginal points and the body points for classification are automatically determined by the DNNSP.

Finally, similar to the transfer learning, the DNNSP learns the common cluster-based features for point clusters of all levels, which makes the features robust, and then the point clusters in different levels are classified by the different fully-connected networks in the DNNSP.

In summary, the main novelty of our learning framework lies in the DMst-based pooling nets, which significantly boosts the performance through utilizing the spatial information. Moreover, the DNNSP can be stacked deeper by means of the pooling nets.

We have performed experiments on airborne laser scanning (ALS) point clouds and terrestrial laser scanning (TLS) point clouds. Experimental results demonstrate that our approach outperforms other methods. Moreover, the advantages between our method and other methods become more obvious in more sophisticated scenes.

2. Related Work

Many recent methods use features such as Spin Images [12], eigenvalues, shape and geometry features [6, 20] in a variety of ways for point cloud classification. Chehata et al. [3] classified point clouds by using the random forests with 21 features which can be categorized into 5 categories and after iterative feature selection, they finally sought out 6 best feature. Guo et al. [10] used JointBoost with 26 features to classify point clouds into 5 classes, such as buildings, vegetation, grounds, electric wires and pylons. Kragh et al. [13] used the SVM classifier with 13 features to classify point clouds and used the different neighborhoods according different point densities. Brodu et al. [2] extracted multi-scale features from different neighborhoods for classifying vegetation, rocks, water and grounds. Zhang et al. [30] clustered point clouds by region growing method and then used the SVM classifier with features of geometry, echoes, radiation degrees and topology of the clusters for point cloud classification. However, these features are sensitive to local geometric noise and they do not capture the global structure of the shape very well [28].

Recently, the deep learning technique can automatically jointly learn the features and the classifiers from data [23]. It has shown flexibility and capability in many applications, like image classification [12], scene Labeling [4] and shape retrieval [32]. The deep learning algorithms, which seek to exploit the unknown structure in the input distribution in order to discover good representations, have been widely applied in 3D object recognition tasks on 3D data like 3D model and RGB-D images. Wu et al. [27] presented volumetric Convolutional Neural Network (CNN) architectures on 3D voxel grids to represent a geometric 3D shape for object classification and retrieval. Zhu et al. [32] used the depth images with different perspectives of 3D objects as input and then used the autoencoder with pre-training by DBN to extract features. Xie et al. [28] used auto-encoder which is imposed the Fisher discrimination criterion on the neurons in the hidden layer for extracting a 3D shape descriptor. Socher et al. [22] used convolutional and recursive neural networks for object reorganization in RGB-D image. There are so few researches for point cloud classification by use deep learning. Guan et al. [9] classify 10 species of trees by using DBN for the vertical profile of the tree point clouds. Based on 2D CNN, [19] proposed a 3D CNN for object binary classification task with LiDAR data. Maturana and Scherer [17] introduced 3D CNNs for landing zone de-

tection from LiDAR data. To tackle a more general object recognition task with LiDAR and RGBD point clouds from different modalities, also study different representations of occupancy and propose techniques. They [18] integrated a volumetric Occupancy Grid representation with a supervised 3D CNN to improve performance. To make 3D CNN architectures fully exploit the power of 3D representations, Qi et al. [21] introduce two distinct network architectures of volumetric CNNs for object classification on 3D data.

The rasterizations in these deep learning methods are used to construct the spatial structure. In this paper, we employ the DMst to determine the spatial structures for the points in the raw point clouds and perform the DMst-based pooling to utilize the spatial structures for point cloud classification.

3. The DNNSP Framework

In the DNNSP framework, there are five kinds of nets Net 1, Net 2, Net 3, Net 4 and Net 5. Net 1 and Net 2 with Net 3 can be stacked to form a deeper architecture. An example of the DNNSP framework is shown in Fig. 1. In this example, there is one Net 1 and one Net 2. The feature of a point is derived from two feature descriptors, and the raw point cloud is clustered into a set of point clusters with three levels. Each sample of the input is the features of all points in a point cluster. Net 1 and Net 2 are the feature representation nets. Net 1 learns the representations from different feature descriptors. After the obtained feature representations are concatenated, Net 2 further learns feature representations from the concatenated ones. Net 3 is a DMst-based pooling net which can utilize the spatial structure of the points in the point clusters. Net 4 contains an average pooling layer in which the features of points are pooled to obtain the cluster-based features. Net 5 is a layer which contains multiple parallel connecting fully-connected networks whose number is the same as the levels of the point clusters.

3.1. Extraction of Cluster-based Features

The features of points can be directly input to a neural network. In this way, it is hard to use the spatial structure among points to achieve high-quality classification results. The clustering operation can provide coarse spatial structures of points and the soft spatial structure can be further obtained from the relations of the points in the clusters. It is noted that the point clusters are not the same sizes, so that the features of points in a cluster cannot be directly concatenated to a fixed-length vector as the input of the neural networks such as CNN or autoencoder. Besides, a point cluster with different point sequences is still the same cluster, but the concatenated feature vectors under different point sequences are not the same.

Based on the above analysis, we first do point-based representations and then pool them to avoid the influences of

Algorithm 1: The DNNSP

For simply, assume there is one Net 1 and Net 2; then the features of a point contain two feature descriptors and finally the point clouds are clustered into three scales.
Input: Training set $\mathbf{X}$ i.e. the clusters with the features of points, the DMst structure of the clusters, the training parameter as learning rate, moment, iterative number, mini-batch size.
Output: Parameters $\mathbf{W}$, $\mathbf{b}$ for each layer
(Initialization): Initialize $\mathbf{W}$, $\mathbf{b}$. Besides,
1. Following the DMst structure, the points are divided into two parts, i.e. the body points and the marginal points. Correspondingly, the $\mathbf{X}_1$ is divided into $\mathbf{X}_1^1$ and $\mathbf{X}_1^2$.
2. As the features of a point contain two feature descriptors, $\mathbf{X}_1^1 = [\mathbf{X}_1^{1,1}, \mathbf{X}_1^{1,2}]$, $\mathbf{X}_1^2 = [\mathbf{X}_2^{2,1}, \mathbf{X}_2^{2,2}]$.
for $t = 1 \dots T$ **do**
(forward propagation):
3. Going through the Net 1.

$$\mathbf{X}_2^{p,f} = \mathbf{X}_1^{p,f}\mathbf{W}_2^{p,f} + \mathbf{b}_2^{p,f} \tag{1}$$

where p is to show a variable belongs to body or margin. f is to show which descriptors a variable connected
4. Going through the Net 3

$$\mathbf{X}_3^{p,f}(i) = \text{maxpooling}(\mathbf{X}_2^{p(\mathbf{\Omega}_i),f}(\mathbf{\Omega}_i)) \tag{2}$$

where i is the ith point in a cluster, $\mathbf{\Omega}_i$ is the set of the point of its subscript.
Because a point can both connect with the body points and the marginal points, p is a function of $\mathbf{\Omega}_i$.
5. Before going through Net 2, the two representations learnt from the two feature descriptors are concatenated as one feature. $\mathbf{X}_3^p = [\mathbf{X}_3^{p,1}, \mathbf{X}_3^{p,2}]$
6. Going through Net 2

$$\mathbf{X}_4^p = \mathbf{X}_3^p\mathbf{W}_4^p + \mathbf{b}_4^p \tag{3}$$

7. Going through Net 3

$$\mathbf{X}_5^p(i) = \text{maxpooling}(\mathbf{X}_4^{p(\mathbf{\Omega}_i)}(\mathbf{\Omega}_i)) \tag{4}$$

8. Going through Net 4

$$\mathbf{X}_6 = \text{averagepooling}([\mathbf{X}_5^1, \mathbf{X}_5^2]) \tag{5}$$

9. Going through Net 5

$$\mathbf{X}_{7,h} = f(\mathbf{X}_6\mathbf{W}_{7.h} + \mathbf{b}_{7,h}) \tag{6}$$

where h shows the hth layer the cluster belonging to.
10. Going through softmax layer

$$y^c = \exp(\mathbf{X}_{7,h}\mathbf{W}_{8.h}^c) / \sum_{c'} \exp(\mathbf{X}_{7,h}\mathbf{W}_{7,h}^{c'}) \tag{7}$$

(back propagation):
11. Back propagation use the cross entropy loss function.

$$L = -\sum_c \tilde{y}^c \log y^c \tag{8}$$

where $\tilde{y}^c$ is the true label
12. Back propagation through all the layers and update the $\mathbf{W}$ and $\mathbf{b}$ in the DNNSP.
Especially, for the Net 3, in step 7

$$\frac{\partial \mathbf{E}}{\partial \mathbf{X}4(i)} = \sum_{j \in \mathbf{\Omega}_i} \frac{\partial \mathbf{E}}{\partial \mathbf{X}_5^{p(j)}(j)} \frac{\partial \mathbf{X}_5^{p(j)}(j)}{\partial \mathbf{X}4(i)} \tag{9}$$

$$= \sum_{j \in \mathbf{\Omega}_i} \frac{\partial \mathbf{E}}{\partial \mathbf{X}_5^{p(j)}(j)} (1[\mathbf{X}_4^{p(i)}(i) = \max \mathbf{X}_4^{p(\mathbf{\Omega}_j)}(\mathbf{\Omega}_j)])$$

in step 4

$$\frac{\partial \mathbf{E}}{\partial \mathbf{X}_2^f(i)} = \sum_{j \in \mathbf{\Omega}_i} \frac{\partial \mathbf{E}}{\partial \mathbf{X}_3^{p(j),f}(j)} \frac{\partial \mathbf{X}_3^{p(j),f}(j)}{\partial \mathbf{X}_2^f(i)} \tag{10}$$

$$= \sum_{j \in \mathbf{\Omega}_i} \frac{\partial \mathbf{E}}{\partial \mathbf{X}_3^{p(j),f}(j)} (1[\mathbf{X}_2^{p(i),f}(i) = \max \mathbf{X}_2^{p(\mathbf{\Omega}_j),f}(\mathbf{\Omega}_j)])$$

where $1[.]$ is the 0/1 indicator.
end
Return: $\mathbf{W}$, $\mathbf{b}$ for each layer

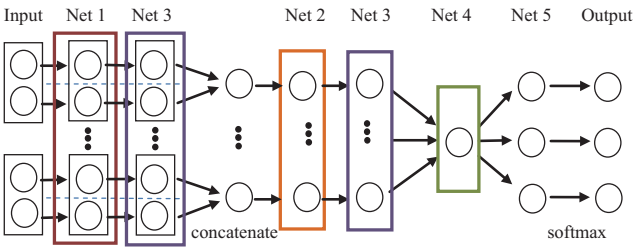

Figure 1. An example of the framework of the DNNSP. The blocks with different color are the different nets. The black circles are feature vectors. The black block at the left of Net 2 is a point-based feature containing two feature descriptors.

the point number and sequences on the classification results. Therefore, each input of the DNNSP is the features of all points in each cluster. In the layers before Net 4, the weights of points are shared during the feature representation. Net 4 is an average pooling layer. After Net 4, the point-based representations are aggregated into the cluster-based features.

The point features are extracted by the method in [19]. It is the 54-dimensional feature vector of each point concatenated by the spin images and the eigenvalue features in three support regions. Afterwards, we construct the multi-level point clusters from the non-ground point cloud. Finally, the features of the points in a point cluster are aggregated as a $54 \times n$ feature matrix, where n is the point number of the cluster. The feature matrix is an input of the DNNSP.

3.2. Point-based Representations

The spin images and eigenvalue feature descriptors describe the point features from different aspects, so the intra relation of each descriptor is closer than the inter relation. Net 1 separately learns the feature representations from the two types of feature descriptors. Steps 2–3 and Eq. (1) in Algorithm 1 show the process.

To express the inter relations of the two feature descriptors, the representations from different feature descriptors are concatenated to one matrix as the input of Net 2. Net 2 further learns the representations. The procedure is described in Steps 5–6 and Eqs. (2), (3) in Algorithm 1. Through Net 1 and Net 2, the DNNSP considers the intra and inter relations of the feature descriptors.

3.3. Utilization of Spatial Structure of Points

A good cluster-based feature can well present the spatial relationships among the points in the cluster. However, the point cloud usually is unorganized. It is essential to first find the spatial layout of the points in the point cloud, and then Net 3 in the DNNSP considers the soft spatial structure information among points.

In the point cloud, the marginal points, which suffer from the scattering of the laser and sometimes are isolated, are diffused. This often makes their features unstable. The points except the marginal points on an object, termed as body points, are even and dense. They generally represent the main structure of the object. Thus, the features extracted from the body points should be highlighted. If the spatial layouts of two different object categories are very similar, these features of the body points can hardly separate them. Therefore, we need more other key points, which are usually on the margins of the objects, to recognize them.

From the above analysis, we should fully consider the contributions of the body points and marginal points to the classification. In the DNNSP, we separate the body points from the marginal points in the point clusters. Then, we configure different weights for them. Finally, to make sure the DNNSP can decide their contributions automatically, the weights of the two types of points can be propagated to each other.

To separate the body and marginal points effectively, the DMst is utilized to organize the points in the non-terrain point cloud. Since the DMst has the advantages of the MST and Dijkstra algorithm, most of the body points are on the trucks and the marginal points are on leaves after the center of a point cluster is taken as the root node.

After the body and marginal points are separated, Net 3 performs the DMst-based pooling operation. Specifically, the max pooling is operated to the representations of a point (i.e. a node in the DMst) and its connected points, and the result of the max pooling is taken as the output of Net 3 of the point. Steps 1, 4, 7 and Eqs. (2), (4) in Algorithm 1 show the process. The different weights keep the features of the body and marginal points from suppressing each other. At the same time, the connection of the leaves and trucks on the DMst keeps the information of the two types of points propagating in the DNNSP by Net 3. It can also be found in the forward propagation equations, i.e. Eqs. (2), (4), and the back propagation equation, i.e. Eqs. (9), (10). The weights of the two types of points are fused, so the DNNSP can automatically determine the contributions of different types of points to the classification. When multiple Net 3 are contained in the DNNSP, the points with different depths on the DMst are mixed with different weights. Therefore, the DNNSP can learn the representations scaled from the whole region to the center of the cluster with the help of Net 3.

3.4. Point Clouds Classification

To learn robust and discriminative features by the DNNSP, similar to the transfer learning, the common features of all levels of each point cluster are learned. Therefore, the structure of the DNNSP before Net 5 is the same for different levels of the point clusters, and point clusters of all levels only learn common features which are output by Net 4. To improve performance of the classification, mul-

tiple fully-connected networks whose number equals to the levels of the point clusters are employed in Net 5. When the multi-level point clusters are input to the DNNSP, the levels of the cluster is recorded. After Net 4, the common features of the clusters only go through the corresponding fully-connected networks. For example, as shown in Fig. 1, the point clusters in the first level enter the top fully-connected networks, and the ones in the second level enter the middle fully-connected networks, and so on. Then the softmax networks are following with the fully-connected networks one by one. The cross entropy loss function is taken as the objective function. Steps 9–10 and Eqs. (6)–(8) in Algorithm 1 show the process.

Finally, the class probabilities of the point clusters of all levels are computed. The labels for points in the point cloud are determined by the point cluster in the finest level with the smallest size. The final class probabilities of a point cluster in the finest level are the multiplication of the class probabilities of the clusters that contain the cluster in all levels. The point clusters are labeled by the maximum class probability.

3.5. Implementation

In the DNNSP, the active function is the min(5, elu(x)). The method in [33] is used for initialization and the Batch Normalization [34] is used after the active function. We applied the stochastic gradient descent to train the DNNSP with mini-batch size of 148. The network learning rate is set to 0.1 and the moment is set to 0.5. In the following experiment, four Net 1 and one Net 2 are used in the DNNSP. Each of Net 1, Net 2 and Net 5 has ten neurons, respectively.

4. Experiment Results

The DNNSP is employed to classify ALS point clouds and TLS point clouds. To evaluate the performance of the DNNSP, we only use the location information, i.e. x, y, z, of each point for the classification.

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 points. The point density is 20–30 points/m^2. The eaves extend outside the building roofs. Due to the scattering, many noisy points are around the eaves, which causes these eaves are hard to be classified to the buildings.

Scene III: The point cloud is Vaihingen dataset provided by International Society for Photogrammetry and Remote Sensing, which are covered by 10 strips. The point clouds are classified to four categories in this paper, i.e. roofs, facades, shrubs and trees, low vegetation. The point density is uneven. The average strip overlap is 30%, and the median point density is 6.7 point/m^2. Point density varies considerably over the whole block depending on the overlap, and in the regions covered by only one strip, the mean point density is 4 point/m^2.

The datasets of Scenes I–III are 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 TLS point clouds provided by Eidgenössische Technische Hoheschule Zürich. The point density is uneven. The point clouds of the scenes are classified into natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artefacts, and cars.

4.2. Classification Results of ALS Point Clouds

For Scenes I and II, the same features, cluster method are used as mentioned in [31]. For Scene III, the point cloud is clustered to multi-level point clusters whose sizes are 60, 120, 240, respectively. The points of the 70% of the clusters belonging to the size level of 240 are randomly selected for training and the other 30% are used for testing.

Table 2 shows the numbers of the points and point clusters of the training and test data in Scenes I–III. Table 3 shows the classification accuracy of the three scenes.

Figs. 2–4 show the training data, test data and classification results of the three scenes. In Figs. 2 and 3, green points are on buildings, blue points are on trees and red points are on cars. In Fig. 4, red points are on roofs, pink points are on facade, blue points are on low vegetation and green points are on shrubs and trees. In Figs. 2–4(d), the gray points are the correct classified points.

As shown in Figs. 3–5, most of the points are classified correctly, which means the DNNSP can extract good cluster features for the categories. In Scenes I and II, only the blue blocks in Figs. 2(c) and 3(c) are misclassified. In the blue block of Fig. 2(c), because of many noises around the eaves, these points look like on a crown. In the blue block of Fig. 3(c), the misclassified points are few, and become a line structure isolated from the building, which may be on an edge of an eave. The car classification accuracy is lower than those of the buildings and trees. The reason is that the car points are not enough and the car features are similar with the features of other categories. In Fig. 5, only some corners of objects are classified wrong since the roofs and the low vegetation are confused. As shown in Fig. 5(d), a part of the true data, the roof and low vegetation points look so similar that only the ridge points on the roofs may distinguish them. The facade points are also few, compared with other categories, but their shapes are not as flexible as the cars in Scenes I and II. Most of facade points are recognized.

Compared with other methods, our method achieves the highest classification accuracy except the tree category in

Scene I	Building	Tree	Car
Training data	37847/1040	70540/2016	5410/173
Test data	201674/5375	218110/6055	7987/249
Scene II	Building	Tree	Car
Training data	64952/1713	39743/1115	4584/142
Test data	157447/4174	74264/2128	7738/239

Scene III	Roof	Shrubs and Trees	Low Vegetation	Façade
Training data	97631/4183	118526/11619	115484/5062	19566/932
Test data	61015/2303	61559/5873	50768/2815	8098/349

Table 1. The number of the points/point clusters in Scene I–III.

Scene I	Building	Tree	Car	Accuracy
Our Method	**97.7/98.8**	**99.2/97.7**	**85.2/98.1**	**98.2**
Method in [10]	89.7/98.1	97.9/89.1	65.2/46.6	92.9
Method in [16]	93.5/96.2	95.3/94.1	75.3/84.6	95.1
Method in [25]	94.0/95.4	95.0/94.3	79.1/60.8	94.5
Method in [31]	95.7/96.2	95.9/95.9	80.8/67.9	95.8
Scene II	Building	Tree	Car	Accuracy
Our Method	**98.9/98.4**	96.2/96.5	**78.4/84.9**	**97.4**
Method in [10]	86.8/91.2	96.8/95.5	44.1/34.8	92.2
Method in [16]	92.7/94.0	95.1/92.6	71.2/65.3	94.3
Method in [25]	90.3/93.9	97.6/96.5	49.4/42.0	94.1
Method in [31]	94.7/94.5	**98.1/97.7**	53.9/60.5	95.5

Scene III	Roof	Shrubs and Trees	Low Vegetation	Facade	Accuracy
Our Method	**97.4/97.8**	**99.8/97.3**	**97.0/99.2**	**95.5/97.3**	**98.0**
Method in [10]	87.5/81.8	81.9/85.2	79.3/82.8	37.4/35.9	81.1
Method in [16]	82.9/89.6	86.6/86.2	91.0/85.1	72.0/65.9	85.9
Method in [25]	79.1/85.5	89.4/86.4	89.6/86.3	53.8/59.6	85.1
Method in [31]	85.9/87.7	89.9/89.2	89.1/87.0	49.8/52.7	86.5

Table 2. The precision/recall (%) and classification accuracy (%) of Scenes I–III

Scene II. As our method can learn more robust feature representation than other methods, the classification accuracy is improved. In Scenes I and II, it is noted that the car accuracy achieved by our method is also higher than those by using other methods, which means our method is competitive for classifying the categories with a few points. In Scene III, many of the roof and facade points are misclassified by other methods. It 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 well using our method, which verifies our method can distinguish the categories even they look similar.

4.3. Classification Results of TLS Point Clouds

For Scenes IV–VI, according to the high point density, the point clouds are clustered to multi-level clusters with the sizes 100, 300 and 500, respectively. We take points of the test. Taking the point clouds of different scenes as the test data can more clearly show the generalization ability of our method. Table 3 lists the number of the points and point clusters in Scenes IV–VI. Table 4 lists the classification accuracy of Scenes V and VI. In Fig. 6, light green points are

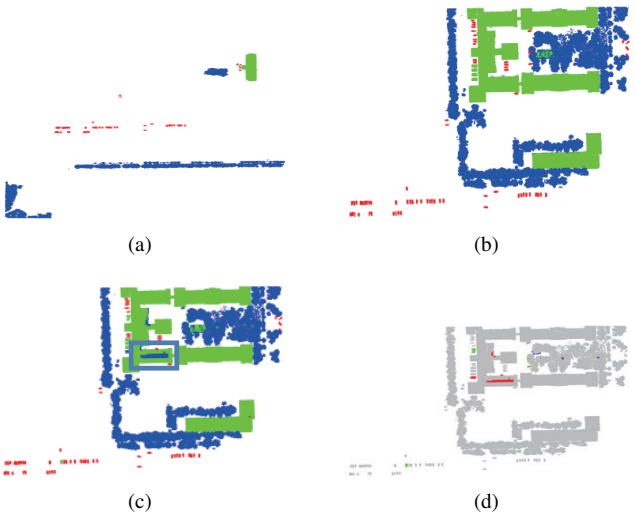

Figure 2. The training data, test data and results of Scene I. (a) The training data. (b) The test data. (c) The classification results. (d) The highlighted wrong points.

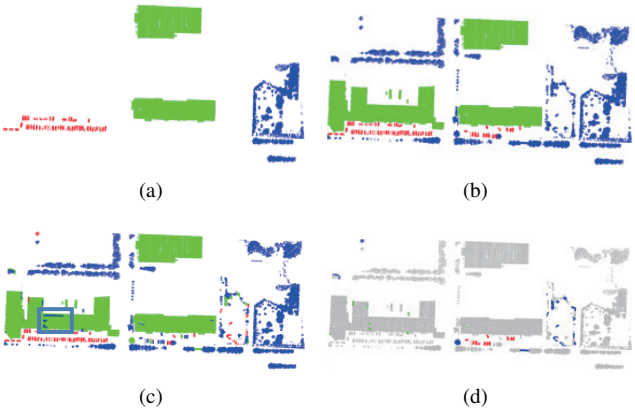

Figure 3. The training data, test data and results of Scene II. (a) The training data. (b) The test data. (c) The classification results. (d) The highlighted wrong points.

on natural terrain, dark green points are on high vegetation, bright green points are on low vegetation, red points are on buildings, purple points are on hard scape, orange points are on the scanning artefacts, and pink points are on cars.

The classification results of Scene V and VI are not as good as those of Scenes I–IV. The shapes of the categories are similar. Especially for the hard scape, they are a class nearly in concept not as the special objects, which contain rocks, fence, steles and so on. Even worse, there are many hard scape points in Scene IV. To fit these points, the DNNSP are overfitting. All the above reasons lead to the low test accuracy although the training accuracy is high. Most of car and low vegetation points are classified into the hard scape and the recall of the hard scape is low. Al-

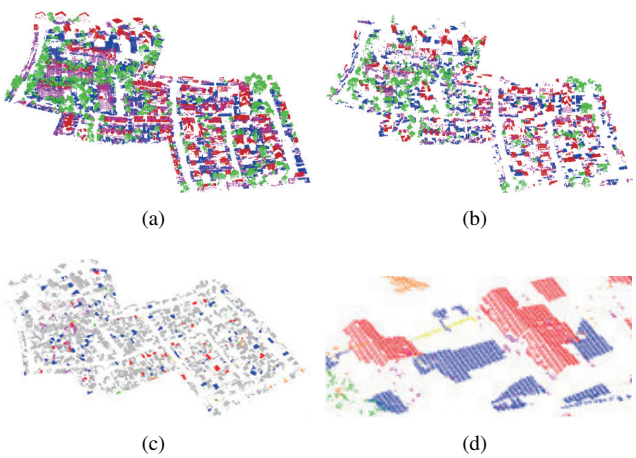

Figure 4. The training data, test data and classification results of Scene I. (a) The training data. (b) The test data. (c) The highlighted misclassified points. (d) The highlighted roof and low vegetation points.

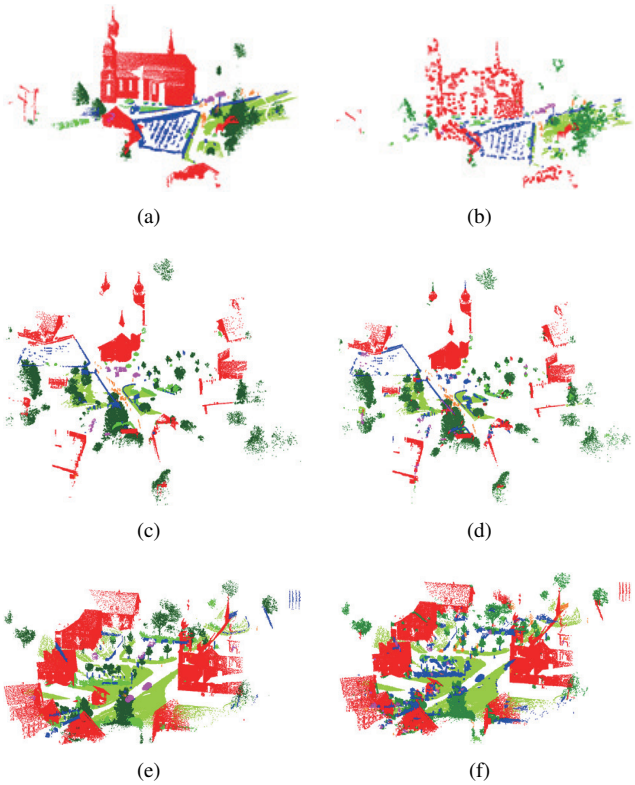

Figure 5. The point clouds of Scenes IV–VI, training data and classification results of Scenes V and VI. (a) Point clouds of Scene IV. (b) The training data in Scene IV. (c) Point cloud of Scene V. (d) The classification result of Scene V. (e) Point cloud of Scene VI. (f) The classification result of Scene VI.

though there are many car points, only three car samples

	Scene IV	Scene V	Scene VI
Natural Terrain	3507576/80268	3174149/72081	4924691/113311
High Vegetation	2537763/59086	1027837/24161	352455/8252
Low Vegetation	49680/1170	592309/13756	172081/3944
Buildings	1241838/28049	539935/12233	1611908/36433
Hard Scape	762982/17190	1260888/28367	46140/1090
Scanning Artefacts	13899/318	7040/176	751/22
Cars	65636/1471	92873/2097	403970/9049

Table 3. The number of the points/point clusters in Scenes IV–VI

Scene V	Our Method	Method in [10]	Method in [16]	Method in [25]	Method in [31]
Natural Terrain	**99.3/97.4**	97.7/76.5	96.7/86.6	96.1/86.4	99.2/98.8
High Vegetation	**94.7/96.0**	83.6/95.2	83.8/95.2	79.7/95.0	89.4/85.7
Low Vegetation	**53.3/89.3**	15.8/48.5	45.8/82.3	45.9/**89.3**	6/56.7
Buildings	**93.9/84.8**	72.4/71.5	72.8/73.6	70.4/69.4	88.9/74.1
Hard Scape	**89.9/78.5**	82.9/71.5	82.1/78.2	80.1/75.8	88.1/66.4
Scanning Artefacts	**88.2/73.5**	37.0/12.9	44.2/17.9	10.5/2.8	52.7/12.9
Cars	11.6/26.1	27.1/68.0	33.7/49.7	**34.9/ 75.4**	13.5/54.2
Accuracy	**91.1**	84.0	84.5	83.6	85.3
Scene VI	Our Method	Method in [10]	Method in [16]	Method in [25]	Method in [31]
Natural Terrain	**98.6/99.5**	76.9/94.7	75.9/92.5	78.2/93.2	84.6/93.7
High Vegetation	**88.1/66.4**	72.1/17.7	78.8/16.3	82.3/14.7	85.4/19.0
Low Vegetation	4.0/31.1	14.8/32.5	**26.5/46.1**	19.1/40.2	16.4/37.5
Buildings	**92.7/98.4**	80.5/79.8	70.1/81.9	65.8/82.4	69.3/83.5
Hard Scape	**66.6**/5.2	28.7/7.4	39.6/**9.6**	32.4/8.7	30.8/7.8
Scanning Artefacts	43.8/2.1	16.3/3.8	30.5/5.6	**59.2/10.7**	44.9/8.0
Cars	1.3/35.1	1.4/14.6	3.6/31.4	**5.2/39.7**	4.3/35.5
Accuracy	**89.3**	71.4	69.0	69.3	74.2

Table 4. The precision/recall (%) and classification accuracy (%) of Scenes V–VI.

are in Scene I, which lead to the training data of cars not enough. Most of car points are labeled as hard scape. The high vegetation and low vegetation are also confused. However, the natural terrain and the buildings are classified correctly, which indicates the accuracy is high if the samples are enough. Meanwhile, our method relies on the input features. In future work, we will utilize the deep neural network directly to extract features from points.

Compared with other methods, our method achieves the highest classification accuracy. Espacially, the accuracy improves at least 20% for Scene VI. It means the DNNSP can learn better feature representations from the point-based features and improve classification accuracy. Moreover, compared to the improvement for ALS point clouds by DNNSP, the improvement is more obvious in the more sophisticated scenes.

4.4. Performance of Nets in the DNNSP

Different numbers of Net 1, Net 2 and Net 3 are used to shown the influences of the nets on the classification. As Net 3 is located behind Net 1 or Net 2, to clear show the

influences of multiple Net3, Net 3 is not used at first and then the classification results with Net 3 are shown.

In Table 5, the DNNSP with 0–4 Net 1 and 0–3 Net 2 are tested. The best classification results are highlighted by bold and the bad classification results are highlighted by italic. For the classification results with/without Net 3, Net 1 is helpful and the best results are all obtained by using at least one Net1; Net 2 is helpless if its number is more than one, and in most cases the more Net 2 are used, the worse the results are. It means separately learning the representations of different descriptors are helpful for classification. When the number of points increases as in Scenes III, V and VI, the Net 2 improves the classification performance, which means the inter relations of the feature descriptors are weak and hard to be learned but still useful for classification. In Scenes I-III, sometimes the classification accuracy has a suddenly decrease as the number of nets changes. This situation does not occur in Scenes V and VI. We argue that the points of Scenes I-III are few and the DNNSP is easy to convergence to the local minimum or overfitting.

Compared with the classification results with/without Net 3, the classification results with Net 3 are more stable than those without Net 3. Also the best results are improved for all the scenes. When the number of Net 2 is determined, the classification accuracy with Net 3 is enhanced with the increase of the number of Net1, but the classification accuracy without Net 3 is random. It is concluded that with the help of Net 3, the DNNSP can be stacked deeper for obtaining better classification performance through more Net 1.

To show the effectiveness of the extracted common feature in the DNNSP, we take the point clusters with the smallest size as the input, and only use one full-connecting network in Net 5. The classification results are listed in Table 6.

From Table 6, we observe that the classification results are obviously lower than those obtained using all levels in Scenes I–III, and the accuracies of the results are about equal in Scenes V and VI. This means Net 5 is helpful for classifying the categories whose points are not enough. For the categories with large points, there is still an increase but slight, which is also similar with the conclusion of the transfer learning.

5. Conclusion

We have presented the DNNSP which can classify point clouds without rasterization. In the DNNSP, the DMst-based pooling utilizes spatial structures of points in the point clouds to make sure point-based representations are discriminative and robust and the DNNSP can be stacked for better classification performance. The experimental results demonstrate the effectiveness of DNNSP for point clouds classification.

Scene I					
Net 2 \ Net 1	Without Net 1	One Net 1	Two Net 1	Three Net 1	Four Net 1
Without Net 2	-	96.7 ± 1.5/ 97.5 ± 0.3	96.9 ± 1.0/ 97.7 ± 0.3	96.9 ± 0.4/ 98.0 ± 0.2	96.9 ± 0.5/ **98.1 ± 0.3**
One Net 2	93.8 ± 2.4/ 90.5 ± 1.3	*86.7 ± 4.5/* *75.3 ± 5.4*	**97.4 ± 0.5/** 94.0 ± 0.9	97.0 ± 0.6/ 95.9 ± 0.4	97.0 ± 0.4/ *85.4 ± 1.4*
Two Net 2	97.7 ± 0.4/ 95.4 ± 0.7	96.7 ± 1.1/ *85.4 ± 6.9*	96.7 ± 0.9/ *80.4 ± 6.1*	97.3 ± 0.6/ 95.2 ± 0.4	96.7 ± 0.7/ 97.1 ± 0.5
Three Net 2	93.2 ± 1.7/ 95.3 ± 1.6	*82.7 ± 6.6/* 96.2 ± 1.0	94.7 ± 0.7/ 96.6 ± 0.8	93.1 ± 2.1/ 97.3 ± 0.5	*88.3 ± 3.6/* 96.4 ± 0.7
Scene II					
Without Net 2	-	95.0 ± 1.9/ 97.3 ± 0.2	93.5 ± 1.4/ 97.3 ± 0.2	93.1 ± 0.8/ 97.8 ± 0.2	92.9 ± 1.8/ **98.0 ± 0.2**
One Net 2	*85.1 ± 4.3/* *89.5 ± 3.8*	**96.5 ± 0.6/** *70.1 ± 9.2*	93.5 ± 1.0/ 94.6 ± 1.1	94.9 ± 0.7/ 96.6 ± 0.5	94.5 ± 0.6/ *88.4 ± 3.3*
Two Net 2	*80.4 ± 5.2/* 96.0 ± 0.5	*85.1 ± 4.7/* *83.2 ± 5.9*	95.7 ± 0.7/ *87.3 ± 4.7*	92.6 ± 1.8/ 97.3 ± 0.4	95.2 ± 0.4/ 96.0 ± 0.4
Three Net 2	*85.7 ± 5.5/* 93.0 ± 2.4	*82.1 ± 4.8/* 94.7 ± 2.3	*86.0 ± 2.3/* 96.0 ± 0.9	*88.0 ± 3.0/* 94.8 ± 2.4	93.6 ± 0.5/ *85.5 ± 5.3*
Scene III					
Without Net 2	-	95.5 ± 0.6/ 94.3 ± 0.9	93.5 ± 0.9/ 94.9 ± 0.5	92.5 ± 1.7/ 95.1 ± 0.4	*85.6 ± 3.1/* 94.6 ± 1.1
One Net 2	93.6 ± 0.2/ 94.5 ± 1.0	*86.3 ± 3.6/* 91.4 ± 1.7	**96.1 ± 0.3/** 94.8 ± 1.2	94.7 ± 0.6/ 95.7 ± 0.7	95.4 ± 0.4/ **96.7 ± 0.7**
Two Net 2	*81.5 ± 5.1/* *82.6 ± 3.4*	95.5 ± 0.5/ 94.5 ± 1.6	*83.8 ± 3.4/* 89.1 ± 1.8	*78.7 ± 3.9/* 91.2 ± 2.0	*89.0 ± 2.5/* 94.1 ± 0.8
Three Net 2	94.0 ± 0.7/ *73.5 ± 7.7*	92.8 ± 0.4/ *87.0 ± 2.8*	94.7 ± 0.6/ 92.5 ± 2.2	91.5 ± 1.8/ 90.3 ± 1.4	*81.2 ± 2.5/* *86.3 ± 5.4*
Scene V					
Without Net 2	-	85.6 ± 2.1/ 85.6 ± 3.2	*78.4 ± 1.7/* *79.4 ± 3.0*	88.6 ± 1.7/ 83.2 ± 3.1	87.9 ± 2.1/ 87.7 ± 1.2
One Net 2	85.3 ± 2.5/ 85.1 ± 2.1	85.5 ± 1.7/ 86.9 ± 1.2	89.7 ± 2.6/ 89.6 ± 1.6	89.0 ± 1.4/ 90.1 ± 1.5	**91.0 ± 1.7/** **91.8 ± 1.7**
Two Net 2	80.8 ± 2.3/ 81.6 ± 3.3	86.5 ± 1.5/ 84.4 ± 4.5	83.7 ± 2.9/ 85.7 ± 2.8	87.4 ± 1.5/ 88.1 ± 1.4	87.4 ± 0.9/ 89.3 ± 1.2
Three Net 2	82.8 ± 1.5/ 82.3 ± 4.9	82.2 ± 2.4/ 83.8 ± 1.1	87.2 ± 2.7/ 84.7 ± 1.6	89.1 ± 1.8/ 86.8 ± 1.4	85.3 ± 3.6/ 87.5 ± 2.9
SceneVI					
Without Net 2	-	86.6 ± 0.4/ 86.5 ± 0.5	86.5 ± 0.3/ 86.5 ± 0.6	87.2 ± 0.5/ 87.2 ± 0.5	87.0 ± 0.8/ 87.8 ± 0.4
One Net 2	84.5 ± 1.3/ 84.8 ± 0.5	86.6 ± 0.6/ 87.5 ± 0.4	**88.1 ± 0.4/** 87.6 ± 0.3	87.5 ± 0.7/ 87.8 ± 0.3	85.9 ± 2.3/ 88.1 ± 0.6
Two Net 2	82.0 ± 2.4/ 84.1 ± 0.6	84.7 ± 1.7/ **89.2 ± 0.8**	87.8 ± 0.3/ 87.1 ± 0.6	86.7 ± 0.6/ 86.6 ± 1.0	86.5 ± 0.8/ 87.0 ± 0.5
Three Net 2	84.8 ± 0.6/ 85.3 ± 0.7	80.4 ± 1.9/ 84.9 ± 1.6	83.0 ± 3.3/ 87.2 ± 0.6	85.0 ± 2.8/ 87.4 ± 0.9	87.6 ± 0.5/ 87.2 ± 0.7

Table 5. The accuracy(%) of classification without/with Net 3.

Scene I	Scene II	Scene III	Scene V	Scene VI
95.6	94.7	92.3	90.3	88.4

Table 6. The classification accuracy (%) obtained by using clusters with one level.

In future work, we will extend the DNNSP to directly extract features from the raw points, and employ the proposed DNNSP to other applications such as 3D object recognition or retrieval.

References

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *PAMI*, 35(8): 1798-1828, 2013. 1

[2] N. Brodu, and D. Lague. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. ISPRS J Photogramm, 68(1):121-134, 2012. 2

[3] N. Chehata, L. Guo, and C. Mallet. Airborne lidar feature selection for urban classification using random forests. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 38, p. W8, 2009. 2

[4] C Farabet, C Couprie, L Najman, and Y Lecun. Learning hierarchical features for scene labeling. PAMI, 35(8):1915-29, 2013 2

[5] A. Frome, D. Huber, and R. Kolluri. Recognizing objects in range data using regional point descriptors. In ECCV, 2004. 1

[6] K. Fukano, and H. Masuda. Detection and classification of pole-like objects from mobile mapping data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 13(2):57-64, 2015. 1, 2

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[8] A. Golovinskiy, V. Kim, and T. Funkhouser. Shape-based recognition of 3D point clouds in urban environments. In ICCV, 2009. 1

[9] H. Guan, Y. Yu, Z. Ji, J. Li, and Q. Zhang. Deep learning-based tree classification using mobile LiDAR data. Remote Sensing Letters, 6(11):864-873, 2015. 2

[10] B. Guo, X. Huang, F. Zhang, and G. Sohn. Classification of airborne laser scanning data using JointBoost. ISPRS J Photogramm, 100:71-83, 2015. 1, 2, 6, 7

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual learning for image recognition. In CVPR, 2016

[12] A. Johnson, and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. PAMI, 21(5):433 – 449, 1999. 2

[13] M. Kragh, R. N. Jørgensen, and H. Pedersen. Object detection and terrain classification in agricultural fields using 3D Lidar data. In ICVS, 2015 1, 2

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[15] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In NIPS, 2011. 1

[16] Z. Li, L. Zhang, X. Tong, B. Du, Y. Wang, L. Zhang, Z. Zhang, H. Liu, J. Mei, X. Xing, and P. Mathiopoulos. A three-step approach for TLS point cloud classification. IEEE Trans. TGRS, 54(9):5412-5424, 2016. 1, 6, 7

[17] D. Maturana, and S. Scherer. 3D convolutional neural networks for landing zone detection from lidar. In ICRA, 2015 1, 2

[18] D. Maturana, and S. Scherer. VoxNet: A 3D convolutional neural network for real-Time object recognition. In IROS, 2015 1, 3

[19] D. Prokhorov. A convolutional learning system for object classification in 3-D lidar data. IEEE Transactions on Neural Networks, 21(5):858–863, 2010. 2, 4

[20] S. Pu, M. Rutzinger, G. Vosselman, and S. Elberink. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS J Photogramm, 66(6):S28-S39, 2011. 2

[21] C. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In CVPR, 2016. 3

[22] R. Socher, B. Huval, B. Bath, C. Manning, and A. Ng. convolutional-recursive deep learning for 3d object classification. In NIPS 2012. 2

[23] A. Stuhlsatz, J. Lippel, and T. Zielke. Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Trans. Neural Netw. Learn. Syst., 23(4):596–608, 2012. 2

[24] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. CNN-RNN: A unified framework for multi-label image classification. In CVPR, 2016

[25] Z. Wang, L. Zhang, T. Fang, P. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li, and D. Chen. A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. IEEE Trans. TGRS, 53(5):2409-2425, 2015. 1, 2, 6, 7

[26] M. Weinmann, S. Urban, S. Hinz, B. Jutzi, and C. Mallet. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Computers & Graphics, 49: 47-57, 2015 1

[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, 2015. 1, 2

[28] J. Xie, Y. Fang, F. Zhu, and E. Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In CVPR, 2015. 2

[29] X. Xiong, D. Munoz, J. Bagnell and M. Hebert. 3-d scene analysis via sequenced predictions over points and regions. In ICRA, 2011 1

[30] J. Zhang, X. Lin, and X. Ning. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. Remote Sensing, 5(8):3749-3775, 2013. 1, 2

[31] Z. Zhang, L. Zhang, X. Tong, Z. Wang, B. Guo, X. Huang, Z. Wang, and Y. Wang. A multilevel point-cluster-based discriminative feature for ALS point cloud classification. IEEE Trans. TGRS, 54(6): 3309-3321, 2016 1, 5, 6, 7

[32] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai. Deep learning representation using autoencoder for 3d shape retrieval. In SPAC, 2014 1, 2

[33] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015. 5

[34] S. Ioffe, and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015. 5