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

Zhen Wang, Liqiang Zhang, Tian Fang, Hao Deng

**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. Recently, the deep learning techniques have produced state-of-the-art results for various computer vision tasks. They have mainly focused on semantic parsing of indoor point clouds by using the rasterization, but large-scale outdoor point cloud classification is still relatively unexplored. To obtain discriminative feature representations from raw outdoor point clouds and avoid losing information in the rasterization process, 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 averagely pools them to cluster features makes sure the DNNSP can directly handle the point clouds. The distance minimum spanning tree (DMst)-based pooling is applied to recognize and describe the spatial information among the points in the point clusters. The pooling can extract the features of points scaled from the whole region to the center of the clusters, which makes the obtained features robust and discriminative. Our method achieves higher classification performance on different types of point clouds than other methods.

***Index Terms***—point cloud classification, deep neural network, weight sharing, special pooling.

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. A number of works have been proposed to study this problem in the past decade. But point clouds classification is still in low progress, comparing with a series of breakthroughs for image classification by the deep learning. Based on the deep learning used, the method can be categorized into two groups. The approaches either directly work on the raw point clouds by fed the features to classifiers [1]-[4]; or use the deep learning for the rasterized point clouds. For the first group of methods, although various hand-designed features [1]-[5] or the hierarchy features extracted from them are used, the features are not as powerful as the feature representations from the deep learning, based on the conclusion in image field. For the second group of methods, various rasterizations are designed to best fit the objects which they study. With the help of rasterization, spatial relationships are constructed for point clouds and the deep learning technology is fit to 3D the rasterization. Most of them force on the indoor applications. 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 design for all the objects with uneven point density and missing data. The rasterization process will 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 [20]. Moreover, augmenting spatial cues with 3D information can enhance object detection in cluttered, realworld environments [3]. 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 classify points by deep learning stems from the inorganization of the point clouds. There is no clearly neighborhood such as the relationship of pixels in images which can be caught by the sliding windows. 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. However, the organization describes the spatial structure which is the most important information for point clouds. As point clouds only supply the position of points, we can not classify a point only by its position but by the shape of the points around it. Therefore, except the rasterization, we propose to proceed another way. We cluster the points to point clusters and describe the clusters by the points in them.

In this paper, a deep neural network with spatial pooling (DNNSP) that exploits rich relational information derived from the scene 3D point cloud is proposed to classify a raw point cloud without rasterization. To avoid rasterization, we do the representations for point features and then do average pooling for them to cluster feature. To utilize the spatial structure, we construct DMst for the points of clusters and do the DMst-based pooling for the points by the organization. At preprocessing, the point clouds are clustered into point clusters, and the feature of each point is calculated. They are taken as the input of the DNNSP. The weight sharing technology is used for points in further feature representation, and the representations of points in clusters are pooled on average to cluster features for classification. In this way, a simple framework is constructed and the framework can be used for point clouds classification without rasterization. However, without the use of the spatial structure, the framework is meaningless, as the most important information of point clouds are not used. Also it will be validated in experiments the simple framework cannot become a deep neural network framework, because the accuracy changes randomly by the increasing of the depth. The distance minimum spanning tree (DMst) algorithm is used to organize point in clusters. Comparing to pooling windows in the images, the connecting points are similar as the pixel in windows and the difference is the number of connecting points are not fixed as windows. Because the DMst can describe the local spatial structure, the DMst-based pooling also inherits this advantage, which makes the representations robust. Pooling helps the feature extracted scaled in image and here it is the same, the representations are scaled from the whole region of the cluster to the center of the cluster. The DMst can also separate the points into marginal points and body points. The body points help to classify the objects with non-resemblance in appearance, and the marginal points help 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 margin points and the body points for classification are automatically determined by the DNNSP. Similar to the transfer learning, the DNNSP learns the common features of all levels in each point cluster, which makes the features robust, and then the point clusters in different levels are classified by the different fully-connected networks in the DNNSP. Finally, the classification results of all levels are combined as the final results.

In summary, the core contribution of this paper is the proposal of the DMst based pooling nets, which can learn features of the point cluster scaled from the whole region to the center of the cluster by utilizing the spatial structure among the points in the point clusters. With the help of the DMst based pooling nets, the DNNSP can be stacked to deep neural network.

We present experiments on three airborne point clouds with the different point density. We show that our method can producing results better than the previous methods. We also used the terrestrial laser scanning (TLS) point clouds for experiments. The point density and the occlusions in TLS point clouds are more serious than airborne point clouds. Compared with the airborne point clouds, the advantages between our method and the previous methods become more obvious in the more sophisticated scenes. Moreover, the results become better when the DNNSP becomes deeper by using the DMst-based pooling nets. All of the evidences show that the DNNSP is an effective deep neural network framework.

**II. RELATED WORK**

Many recent methods use hand-designed features combining with classifiers for point cloud classification. Various features such as Spin Images [21], eigenvalues [22] or specific color, shape and geometry features [23] have been used in a variety of ways. Chehata et al. [24] 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. [25] used JointBoost with 26 features to classify point clouds into 5 classes, such as buildings, vegetation, grounds, electric wires and pylons. Kragh et al. [26] used the SVM classifier with 13 features to classify point clouds and used the different neighborhoods according different point densities. Brodu et al. [27] extracted mulit-scale features from different neighborhoods for classifying vegetation, rocks, water and grounds. Zhang et al. [27] clustered point clouds by region growing method and then used SVM with features of geometry, echoes, radiation degrees and topology of the clusters to point cloud classification.

Recently, the deep learning technique can automatically jointly learn the features and the classifiers from data [13]-[16]. It has shown flexibility and capability in many applications, like image classification [17], scene Labeling [18] and shape retrieval [19]. 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 RGB-D images and point clouds. Wu et al. [29] presented volumetric CNN architectures on 3D voxel grids to represent a geometric 3D shape for object classification and retrieval. Zhu et al.[30] 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. Their method obtains good performance, but through augment the 2D CNN with pre-training from ImageNet RGB data. Su et al. [31] evaluated that training a CNN on multiple 2D views achieves a significantly higher performance. Xie et al. [19] 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. [32] 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. [33] classify 10 species of trees by using DBN for the vertical profile of the tree point clouds. Based on 2D convolutional neural network, [34] proposed a 3D CNN for object binary classification task with LiDAR data. Maturana and Scherer [35] introduced 3D CNNs for landing zone detection 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 [14] 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. [15] 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. Therefore, the key to apply the deep learning to the raw point clouds is explicating spatial structure for raw point clouds and generating neural network based on it. In this paper, we use the DMst to construct the spatial structure for the raw point clouds and propose the DMst-based pooling to utilize the spatial structure for point clouds classification.

Zhuqiang Li, Liqiang Zhang, Xiaohua Tong, Bo Du, Yuebin Wang, Liang Zhang, Zhenxin Zhang, Hao Liu, Jie Mei, Xiaoyue Xing, 2016. A Three-step Approach for TLS Point Cloud Classification. [IEEE Transactions on](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=36" \t "_blank) Geoscience and Remote Sensing, 54(9), 5412-5424. Sept. 2016.

Martinovic, A., Knopp, J., Riemenschneider, H. and Van Gool, L., 2015. 3D all the way: Semantic segmentation of urban scenes from start to end in 3d. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4456-4465.

1. A. Golovinskiy, V. G. Kim, and T. Funkhouser, “Shape-based recognition of 3D point clouds in urban environments,” ICCV, 2009.
2. [12] A. Teichman, J. Levinson, and S. Thrun, “Towards 3D object recognition via classification of arbitrary object tracks,” in ICRA, 2011, pp. 4034–4041.

[5] A. Frome, D. Huber, and R. Kolluri, “Recognizing objects in range data using regional point descriptors,” ECCV, vol. 1, pp. 1–14, 2004.

[6] J. Behley, V. Steinhage, and A. B. Cremers, “Performance of histogram descriptors for the classification of 3D laser range data in urban environments,” in ICRA, 2012, pp. 4391–4398.

[7]Li Shen, Multi-Level Discriminative Dictionary Learning With Application to Large Scale Image Classification

[8]G. Griffin and P. Perona, rona, iffin and P. Perona, iminative Dictionary Learning With Application to Large Scale Image Classific[9] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei, i-Fei, Fei-Fei, ei, . semantivisual image hierarchy,ge hierarchy,minative Dictionary Learning With

[10] T. Gao and D. Koller, ller, Gao and D. Koller, hy,minative Dictionary Learning With Application to Large Scale Image ClICCV, Nov. 2011, pp. 20720727220[11] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, nenbaum, nenbaum, enbaum, aum, . Tenenbaum, enbaum, ning With Application to Large Scale Image ClICCV, Nov. 2011, pp

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, 2015.

[14] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition.

[15] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2016.

[16] A. Stuhlsatz, J. Lippel, and T. Zielke, “Feature extraction with deep neural networks by a generalized discriminant analysis,” IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 4, pp. 596–608, Apr. 2012.

[17] Wang jiang. CNN-RNN: A Unified Framework for Multi-label Image Classification

[18] Cle´ment Farabet, Learning Hierarchical Features for Scene Labeling

[19] Jin Xie, Yi Fangy, Fan Zhuy, and Edward Wong. DeepShape: Deep Learned Shape Descriptor for 3D Shape Matching and Retrieval

[20] H.S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor

scenes. In NIPS, 2011.

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

[22] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, et al., "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," Geoscience and Remote Sensing, IEEE Transactions on, vol. 53, pp. 2409-2425, 2015.

[23] And¯elo Martinovic. 3D All TheWay: Semantic Segmentation of Urban Scenes From Start to End in 3D.

[24] 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.

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

[26] M. Kragh, R. N. Jørgensen, and H. Pedersen, "Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data," in Computer Vision Systems, ed: Springer, 2015, pp. 188-197.

[27] N. Brodu and D. Lague, "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, vol. 68, pp. 121-134, 2012.

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

[29] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, et al., "3d shapenets: A deep representation for volumetric shapes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912-1920.

[30] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3d shape retrieval," in Security, Pattern Analysis, and Cybernetics (SPAC), 2014 International Conference on, 2014, pp. 279-284.

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

[32] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in Advances in Neural Information Processing Systems, 2012, pp. 665-673.

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

[34] D. Prokhorov, “A convolutional learning system for object classification in 3-D lidar data,” IEEE TNN, vol. 21, no. 5, pp. 858–863, May 2010.

[35] D. Maturana and S. Scherer, "3D convolutional neural networks for landing zone detection from lidar," in Robotics and Automation (ICRA), 2015 IEEE International Conference on, 2015, pp. 3471-3478.

**III. The DNNSP Framework**

In the DNNSP framework, there are five kinds of nets, such as Net 1, Net2, Net3, Net4 and Net5, and the number of Net 1 and Net 2 can be adjusted according to different point cloud data. An example of the DNNSP framework is shown in Fig.1. In this example, there is one Net 1 and one Net 2；the features of a point contain two feature descriptors, and the raw point clouds are clustered into three levels. Each sample of the input is the features of all points in a point cluster. Net1 and Net 2 are the feature representation nets. Net 1 learns the features from different types of feature descriptors. After the features representedfrom different types of feature descriptors are concatenated, Net 2 further learns the features from the concatenated features.Net 3 is a DMst-based pooling net which can utilize the spatial structure of the points in clusters. Net 4 contains an average pooling layer. The features of points are pooled to obtain the cluster-based features through this layer. 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.

Net4

Net5

Concatenate

Net 1

Net 2

softmax

input

output

Net3

Net3

Fig.1. An example of the framework of the DNNSP. The block with different color are the different nets. The black circles are feature vectors. The black block at the left of Net2 is a point-based feature containing two feature descriptors.

|  |
| --- |
| Algorithm 1. DNNSP |
| For simply, assume there is one Net1 and Net2; then the features of a point contain two feature describers and finally the point clouds are clustered into three scales.  **Input**: Training set **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 **W**, **b** for each layer  **(Initialization):** Initialize **W, 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 X1 is divided into X11 and X12.  2, As the features of a point contain two feature describers, the X11=[X11,1, X11,2 ]，X12=[X22,1, X22,2].  **For** t =1…T **do**  **(forward propagation):**  3, Going through the Net1.  (1)  where *p* is to show a variable belongs to body or margin. f is to show which describer a variable connected  4， Going through the Net3  (2)  wherei is the ith point in a cluster,  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 .  5，Before going through the Net2, the two representations learnt from the two feature describers are concatenated as one feature.  6，Going through the Net2  (3)  7，Going through the Net3  (4)  8，Going through the Net4  (5)  9，Going through the Net5  (6)  whereh shows the hth layer the cluster belonging to.  10，Going through softmax layer  (7)  **(back propagation):**  11. Back propagation use the cross entropy loss function  (8)  where  is the true label  12. Back propagation through all the layers and update the **W** and **b** in the DNNSP.  Especially, for the Net3, in step 7  (9)  in step 4  (10)  1[.]is the 0/1 indicator。  **end**  **Return: W**, **b** for each layer |

A. Cluster Features Extraction

The features of points can be directly input to the DNNSP. In this way, it is hard to use the spatial structure among points to achieve high-quality classification results. Clustering operation can provide coarse spatial structure of points and the soft spatial structure can be further obtained from the connection of the points in clusters. Therefore, the raw point cloud is first segmented into point clusters, and then the features of points in a cluster are fused into a cluster feature. It is noted that the point clusters are not the same sizes, so that the features of points in a cluster cannot directly concatenated to a vector with fixed-length as the input of the neural networks such as CNN or autoencoder. Besides，a cluster with a different points sequence is still the same cluster but the input vector will not be the same one, if any changes have been done in the vector. We pool the representations from features of points in the clusters to avoid the influences of 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 feature representation. Net 4 is an average pooling layer. After Net 4, the representations from features of points in a cluster are aggregated into the feature of the cluster.

The input of the DNNSP is obtained by the following step. The features of points extracted as the method in [18]. As a first step, the ground points are removed from the point clouds. The removal of the ground points helps to determine the connectivity of objects. Next, the 54-dimensional features of each point are obtained, which is 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 points in a cluster are concatenated as a feature matrix with 57×*n*, where *n* is the point number of the cluster. The feature matrix is an input of the DNNSP.

After the last Net3, the representations are pooled by Net 4. The output of Net 4 can be taken as a cluster feature with the same dimension as the output of the last Net3.

**B. Learning the intra and inter relations of the feature** describer**s**

The spin images and eigenvalue features describe the point feature from different viewpoints, so the intra relation of each describer**s** is closer than the inter relation of the two. Two nets are configured to separately handle the two types of featuredescriber**s**. Steps 2-3 and Eq. (1) in Table 1 show the process.

To show the inter relations of the two featuredescriber**s**, the representations from different feature describer**s** are concatenated to one matrix as the input of Net2. Net2 further learns the representations. Steps 5-6 and Eqs. (2)-(3) in Table 1 show the process. Through Net 1 and Net 2, the DNNSP considers the intra and inter relations of the feature describer**s**.

**C. Utilization of spatial structure information of points**

A good point cluster feature can well describe the spatial and topological relationships among the points in the cluster. Point clouds usually are unorganized and there are not definitely adjacent relationships among neighboring points. 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 clouds, the marginal points which suffer from the scattering of the laser and sometimes are isolate are diffused, which make their features unstable. However, the points except the marginal points on an object, termed as body points, are even and dense, and they generally represent the main structure of the object, Thus, the features extracted from the body points should be highlight. However, if two different object classes are very similar, these features can hardly separate them. We need more 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 body points and the marginal points for the classification. In the DNNSP, we first separate the body points and marginal points. 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 propagate to each other.

To separate the body and marginal points effectively, the DMst [] is utilized to organize the points in the non-ground point cloud. Since the DMst has the advantages of the 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 does the DMst-based pooling operation. Specifically, the max pooling is operated to the representation 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. In the DNNSP, Net 3 is following with Net 1 or Net 2. Steps 1, 4, 7 and Eqs. (2), (4) in Table 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 the trucks 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 decide the contributions of the points to the classification. When many Net 3 are used in the DNNSP, the points with different depths in the DMst are mixed with different weights. Therefore, the DNNSP can learn representations scaled from the whole region of the cluster to the center of the cluster with the help of Net 3.

**D. Point clouds Classification**

As the point density of the point cloud is uneven, the features of point clusters would be greatly different, even belonging to the same class. 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 all levels of point clusters only learn common features which are output by Net 4. To improve performance of the classification for the different levels of the point clusters, the fully-connected networks with the same number as the levels of point clusters are used in Net5. 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 cluster in the first level enters the top fully-connected networks, and the point cluster in the second level enters the middle fully-connected networks, and so on. The softmax layer is applied for classifying the point clusters. The softmax networks are following with the fully-connected networks one by one. The cross entropy loss function is used as the objective function. Steps 9-10 and Eqs. (6)-(8) in Table 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 point clusters in the finest level are the multiplication of class probabilities of the clusters who contain the clusters in other levels. Finally, the point clusters are labeled by the maximum class probability.

**E. Implementation**

In the DNNSP, the active function is the min(5, elu(x)), in which the upper limit is set to 5. The aim is to prevent the DNNSP from making mistake caused by too large values of neurons. The initialization using the method in [32] and the Batch Normalization [33] is used after the active function. The stochastic gradient descent is applied to train the DNNSP with the batch 148. The network learning rate is set to 0.1 and the moment is set to 0.5. In the DNNSP, 2 Net1 and 1 Net2 are used, and 10 neurons are used in Net 1, Net 2 and Net 5.

**III. EXPERIMENTAL RESULTS**

To verification the performance of DNNSP for the point cloud classification, the DNNSP is used to classify the airbone point clouds and the ground point clouds. As we want to mine the ability for classification by the point clouds, only the points with their positions x, y, z are used, although some datasets supply some other features such as RGB, intensity and so on.

1. **Dataset**

Six scenes are used for the verification.

Scene I and II: The point clouds of the two scenes in Tianjin contain buildings, trees and a few cars points, with the point density 20–30 points/m2. The eaves extend outside the roof and because of scattering, many noise are around the eaves, which causes these eaves are hard to be classified to buildings.

Scene III: The point clouds are the Vaihingen dataset supplied by ISPRS, which are covered by 10 strips. The point clouds are classified to four classes in the paper, such as 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/m2. Point density varies considerably over the whole block depending on the overlap, but in the regions covered by only one strip, the mean point density is 4 point/m2.

The above data is airborne 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.

Scene IV – VI: They are parts of the dataset which supplied by Eidgenössische Technische Hoheschule Zürich. As they are ground point clouds, the point density is uneven. The point clouds of the scenes are classified to natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artifacts, and cars.

**B．Classification results of airborne point clouds**

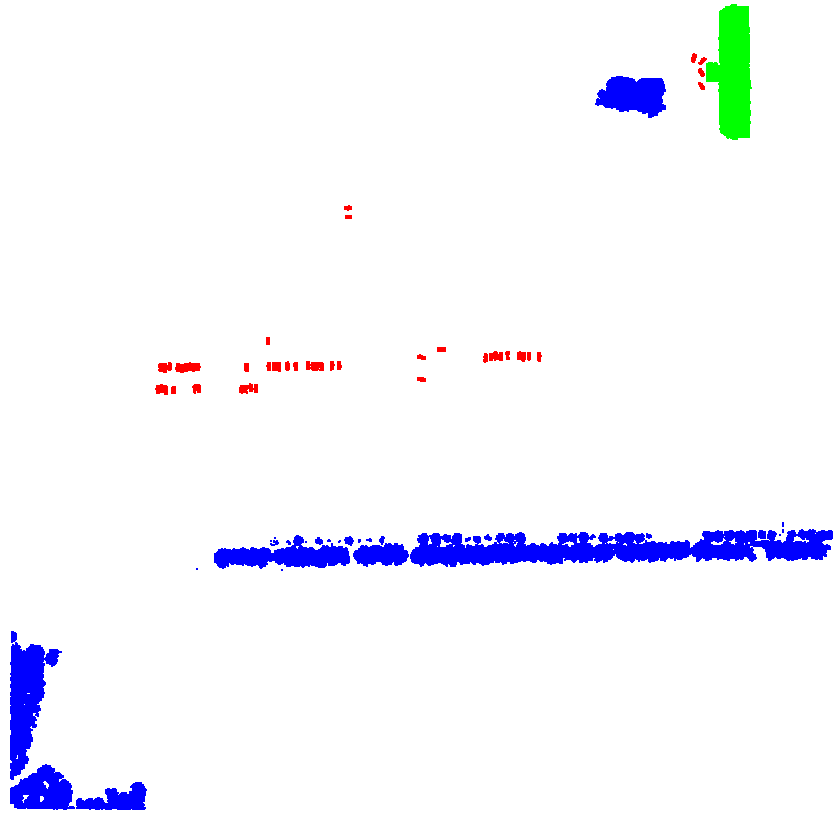
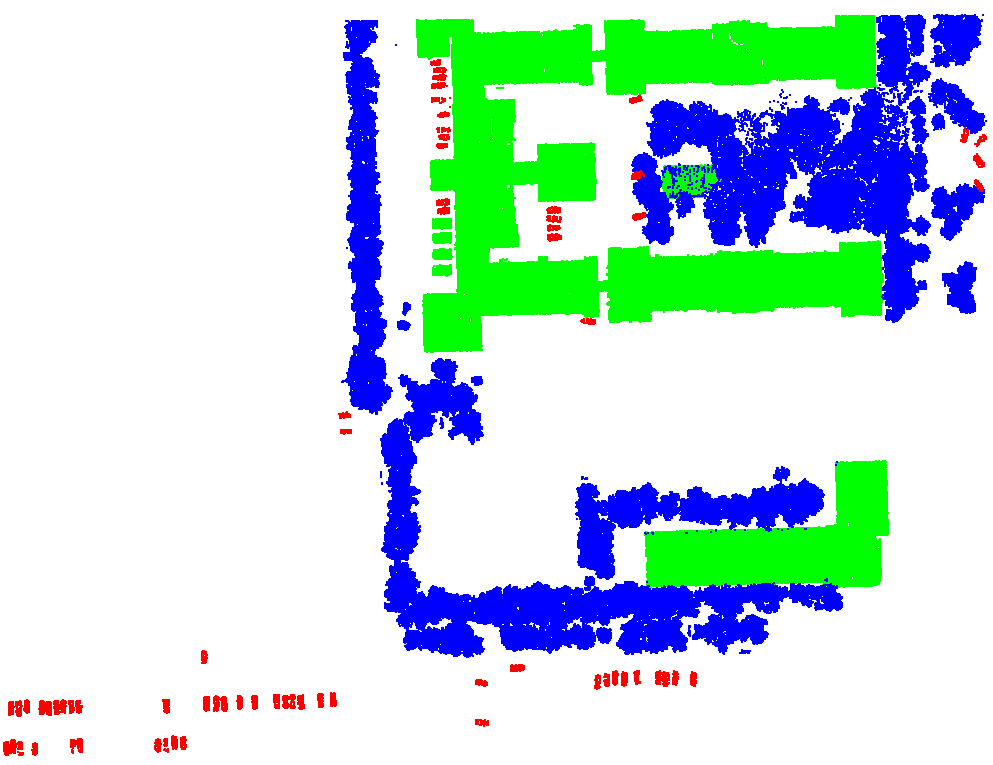
For scene I and II, the same features, cluster method, training and test data are used as mentioned in [18]. For scene III, point clouds are clustered to multi-level clusters with the sizes 60,120,240. The points of the 70% of the clusters belonging to the size level of 240 are randomly selected for training and the other 30% are used for testing..

Table 2 shows the number of the point and point cluster in the training and testing data for scene I-III. Table 3 shows the accuracy of the classification of the three scenes. Fig. 2-4 shows the training data, testing data and results of the three scenes. In the Fig. 2 and 3, green points are buildings, blue points are trees and red points are cars. In Fig. 4, red points are roofs, pink points are facade, blue points are low vegetation and green points are shrubs and trees. In Fig.2-4 (d), the gray points are the points classified right.

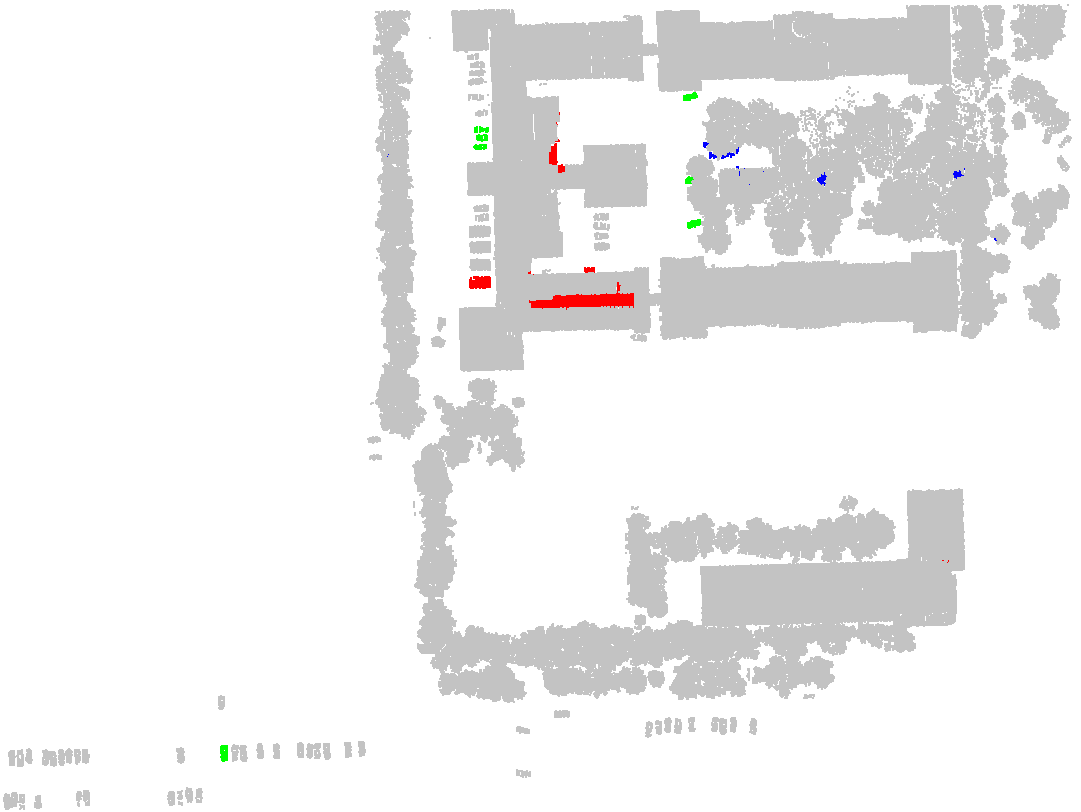
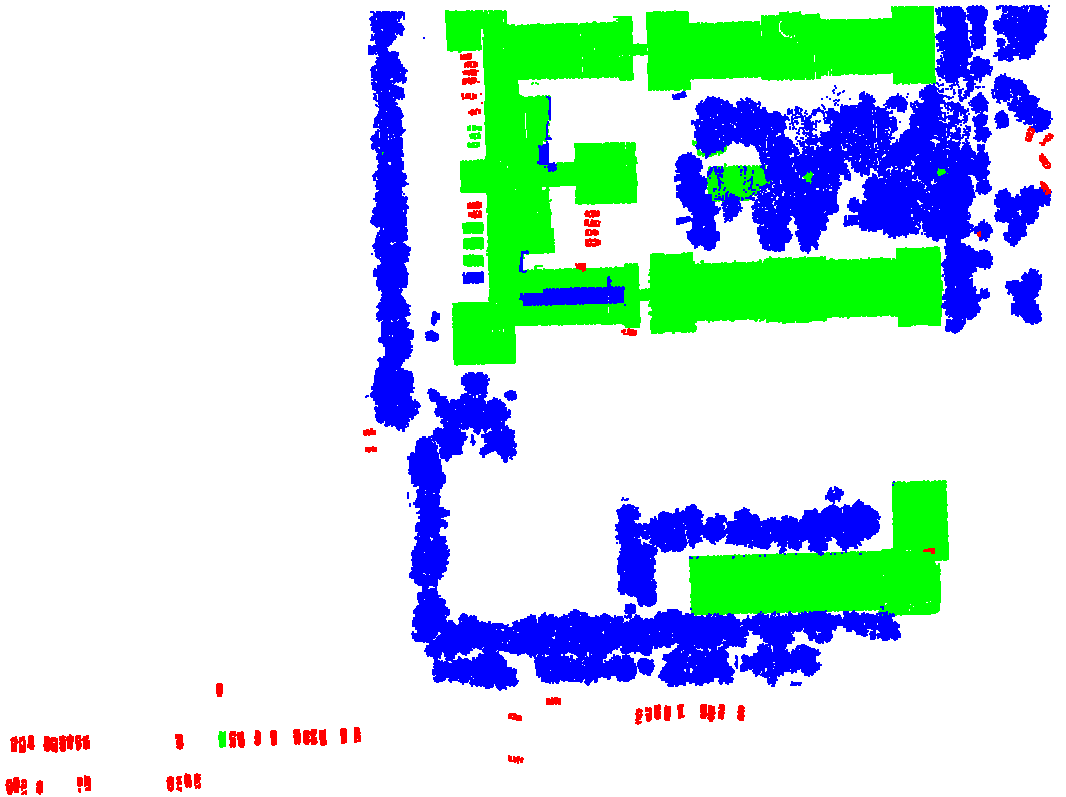
Table 2 The number of the point and point cluster in scene I-III

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Facade |
| Training data | 97631(4183) | 118526(11619) | 115484(5062) | 19566(932) |
| Test data | 61015(2303) | 61559(5873) | 50768(2815) | 8098(349) |

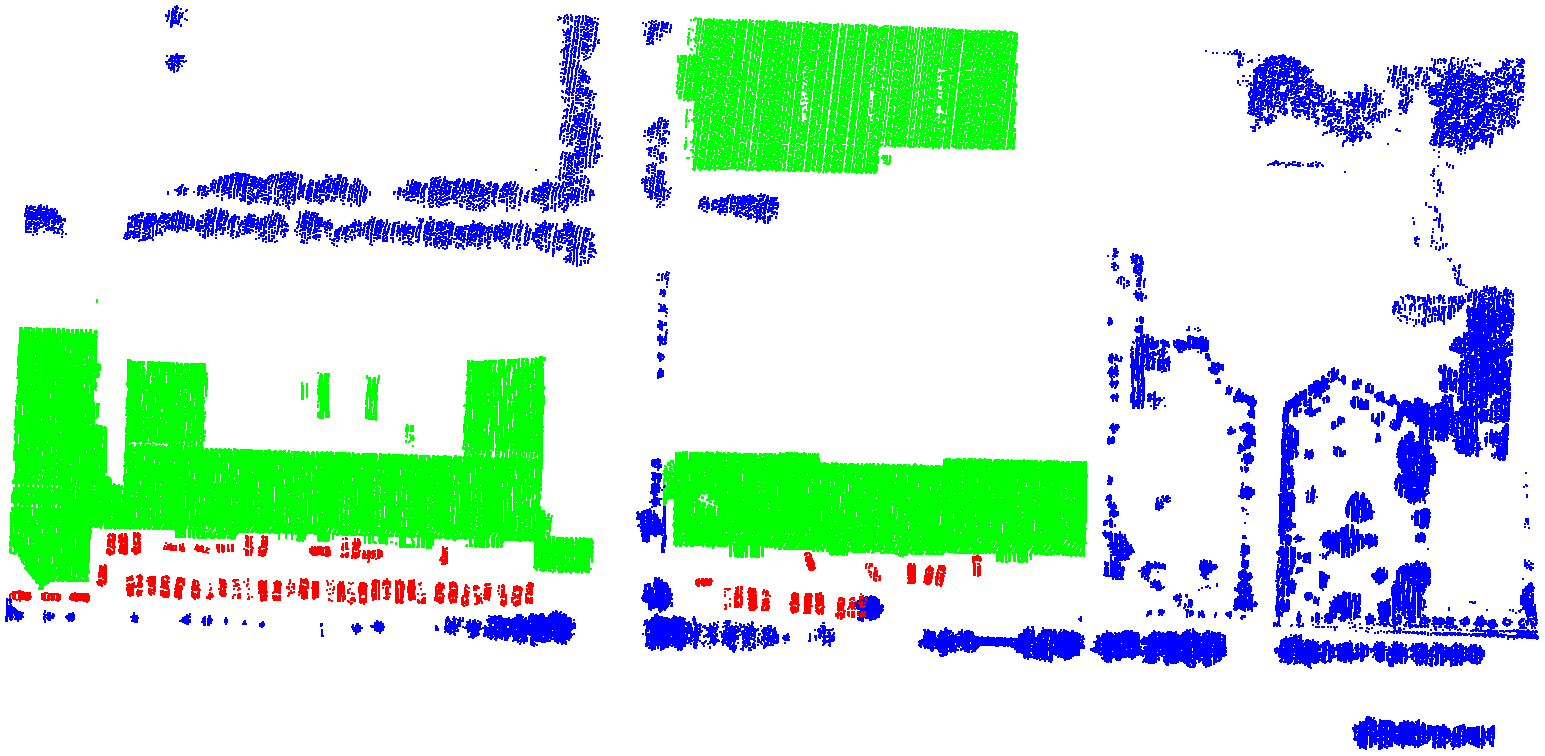
 

(a)training data (b) testing data

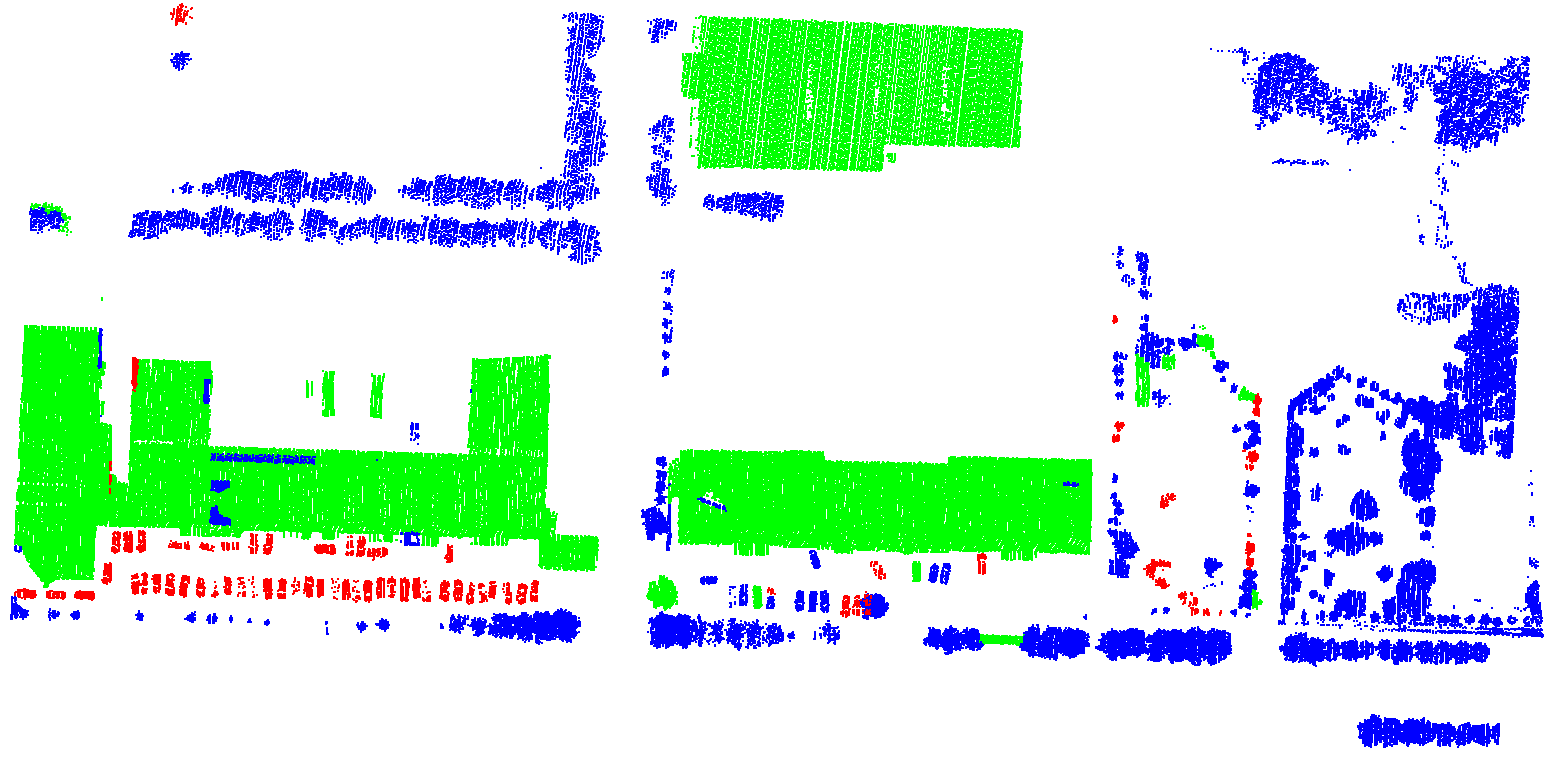
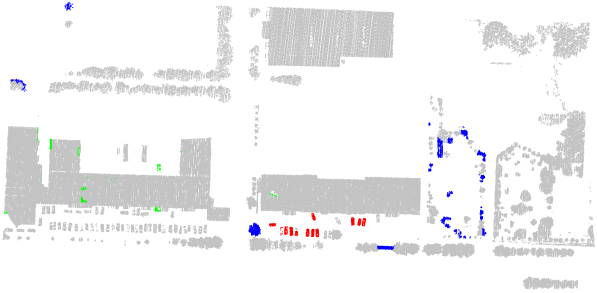


(c) classification result (d) highlight wrong points

Fig.2 the training data, testing data and results of scene I

(a)training data (b) testing data

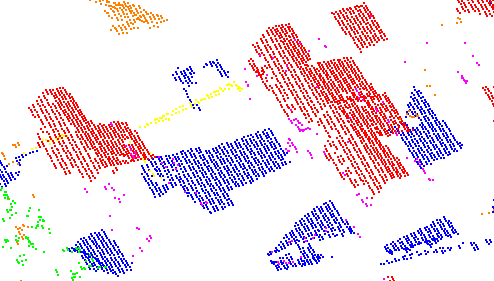
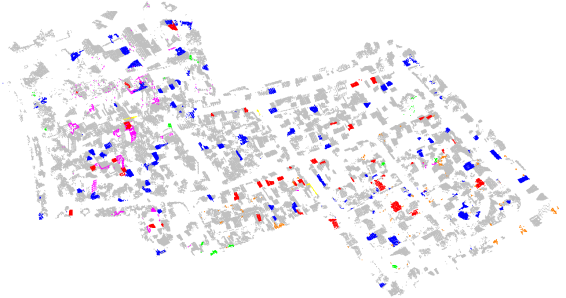
 

(c) classification result (d) highlight wrong points

Fig.3 the training data, testing data and results of scene II



(a) training data (b) testing data



(c) highlight wrong points (d) highlight roof and low vegetation points

Fig.4 the training data, testing data and results of scene II

Table 3 the precision/recall and accuracy of the classification results of 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[17] | | 94.0/95.4 | | 95.0/94.3 | | 79.1/60.8 | | 94.5 | |
| Method in[18] | | 95.7/96.2 | | 95.9/95.9 | | 80.8/67.9 | | 95.8 | |
| Method in[4] | | 89.7/98.1 | | 97.9/89.1 | | 65.2/46.6 | | 92.9 | |
| Method in[35] | | 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 in[17] | | 90.3/93.9 | | 97.6/96.5 | | 49.4/42.0 | | 94.1 | |
| Method in[18] | | 94.7/94.5 | | **98.1/97.7** | | 53.9/60.5 | | 95.5 | |
| Method in[4] | | 86.8/91.2 | | 96.8/95.5 | | 44.1/34.8 | | 92.2 | |
| Method in[35] | | 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 | **97.4/97.8** | | **99.8%/97.3** | | **97.0%/99.2** | | **95.5/97.3** | | **98.0** | |
| Method in[17] | 79.1/85.5 | | 89.4/86.4 | | 89.6/86.3 | | 53.8/59.6 | | 85.1 | |
| Method in[18] | 85.9/87.7 | | 89.9/89.2 | | 89.1/87.0 | | 49.8/52.7 | | 86.5 | |
| Method in[4] | 87.5/81.8 | | 81.9/85.2 | | 79.3/82.8 | | 37.4/35.9 | | 81.1 | |
| Method in[35] | 82.9/89.6 | | 86.6/86.2 | | 91.0/85.1 | | 72.0/65.9 | | 85.9 | |

As the results shown in Fig.3-5, almost all the points are classified well, which shows the DNNSP can extract good cluster features for these classes. In scene I and II, only the blue blocks in Figs. 2 and 3(c) are classified wrong. In the blue block of Fig.3, because of so much noise around the eaves, these points look like on a crown. In the blue block of Fig.4, the points classified wrong are so few and become a line structure isolated outside the building, which are maybe on an edge of an eave. The accuracy of car classification is not so high as buildings and trees. This is maybe because the car points are not enough and the features of car are similar with the features of the others classes. In Fig.5, only some corners of objects are wrong, and most of the wrong points are because the roof and the low vegetation are confused. As shown in Fig.5(d), a part of the true data, the roof and the low vegetation points look so similar only the ridge points on the roofs may distinguish them. The facade points are also few comparing with other classes, but their shapes are not as flexible as the cars in scene I and II. Most of facade points are recognized.

Compared with the other methods, our method outperforms all the methods on all labels except the tree label in scene II. As our method can learn deeper feature representation than other method, the classification accuracies are improved. In scene II, although the performance of trees is not as good as the method in [17] and [18], the performance of other labels especially the car label is obviously better. In scene I and II, it is surprising that the car accuracy is also higher than other method, which means our method is competitive for the classes with a few points. In scene III, many of the roof and facade points are classified wrong by other methods. This 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 classes are all classified well by us, which shows our method can distinguish the classes even they look similar.

**B．Classification results of ground point clouds**

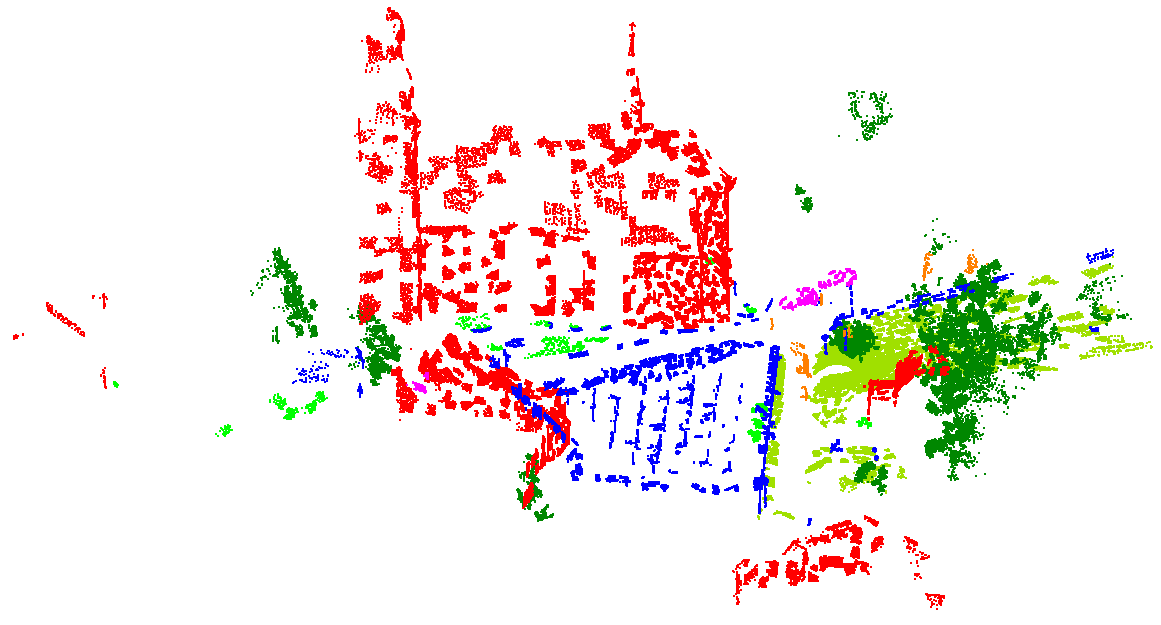
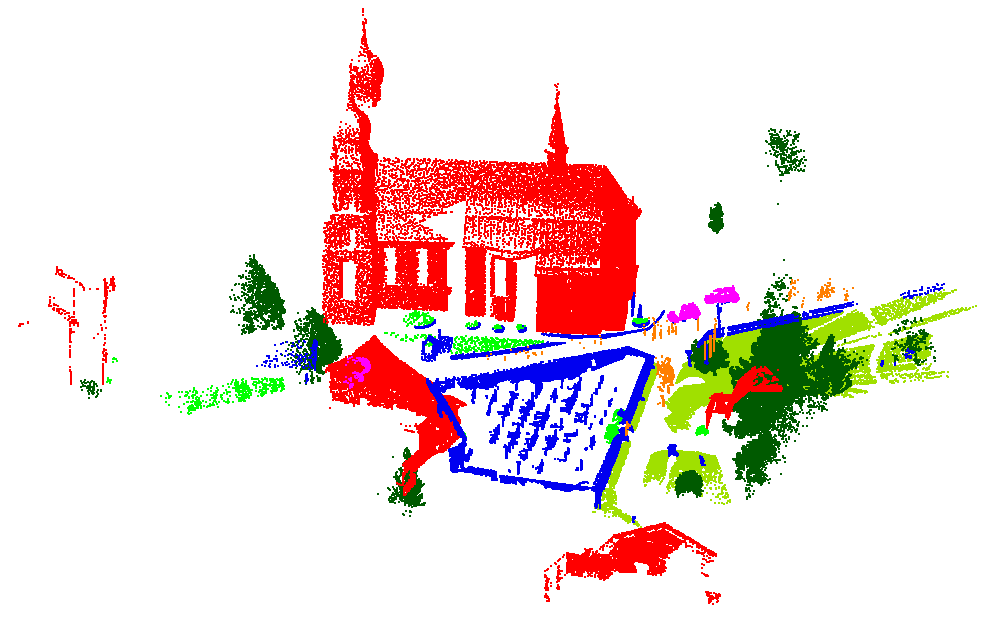
For scene IV-VI, the point clouds are clustered to multi-level clusters with the sizes 60,120,240. Points clouds of 33% of the clusters belonging to the size level of 240 in scene IV are used for training and scene V and VI are used for testing. Using the different scenes as testing data can show the generalization ability of our method more clearly. Table 4 shows the number of the point and point cluster in scene IV-VI. Table 5 shows the accuracy of the classification of scenes V and VI. Fig. 6 shows the scenes IV-VI, the training data and the classification results of scene V and VI. In Fig. 6, light green is natural terrain points, dark green is high vegetation points, bright green is low vegetation points, red is buildings points, purple is hard scape, orange is scanning artifacts points, pink is cars.

Table 4 The number of the point and point cluster in scene IV-VI

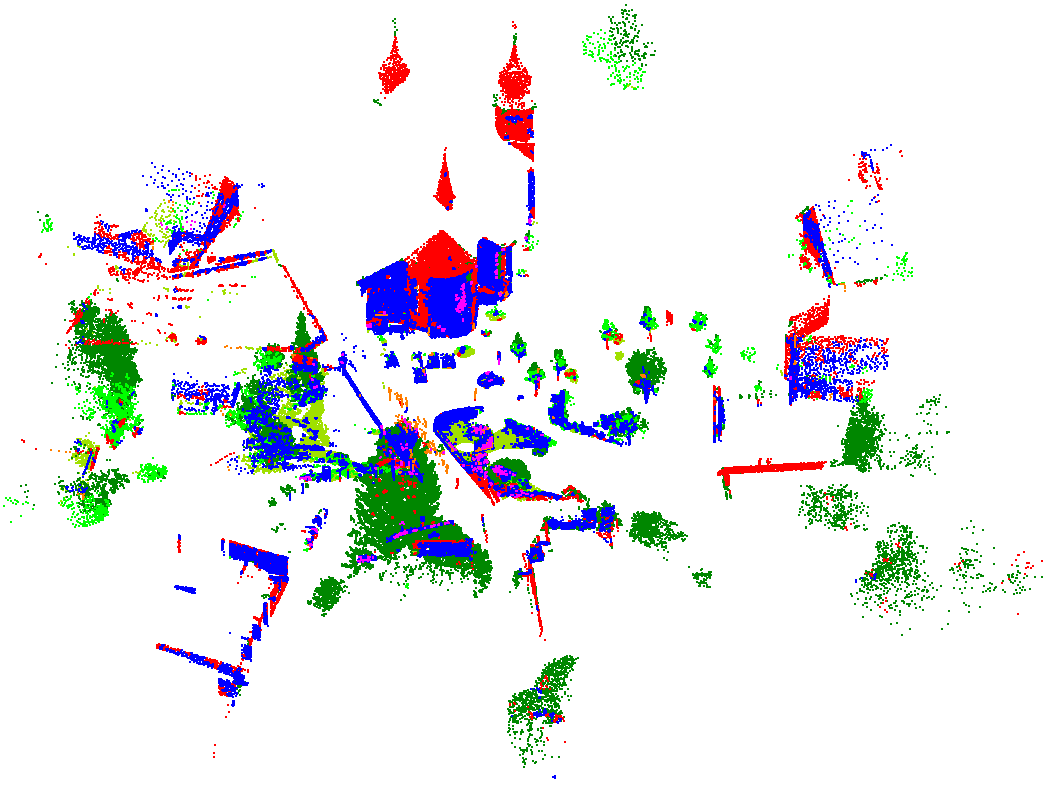
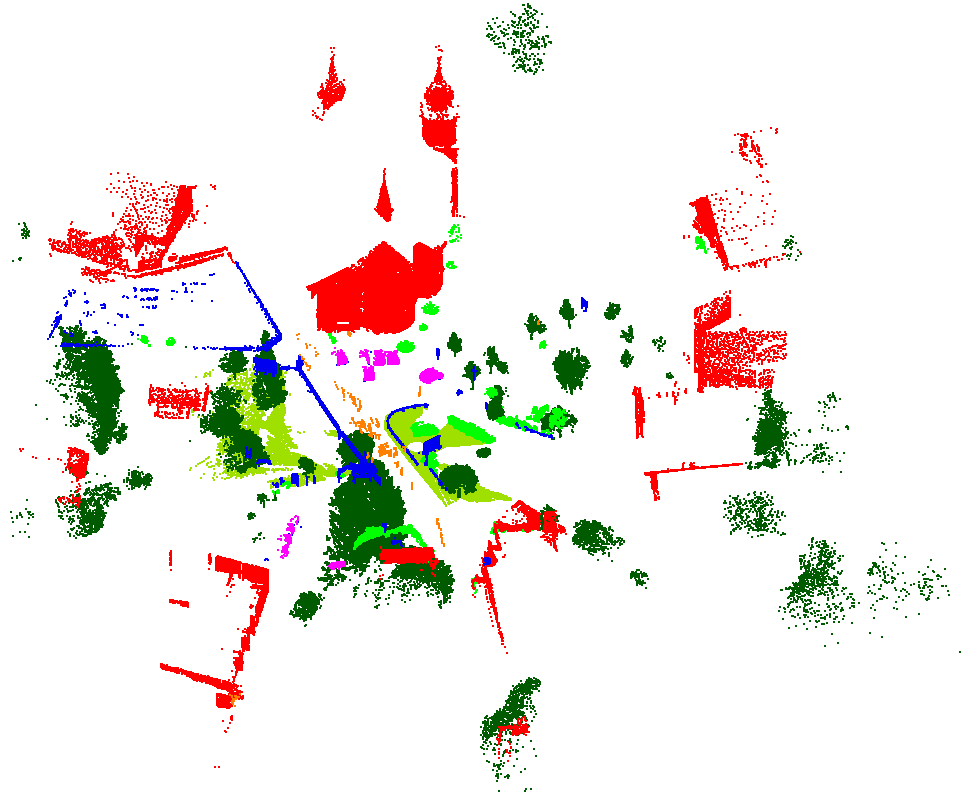
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | natural terrain | high vegetation | low vegetation | buildings | hard scape | scanning artifacts | cars |
| Scene IV | 3174149  (72081) | 1027837  (24161) | 592309  (13756) | 539935  (12233) | 1260888  (28367) | 7040  (176) | 92873  (2097) |
| Scene V | 3507576  (80268) | 2537763  (59086) | 49680  (1170) | 1241838  (28049) | 762982  (17190) | 13899  (318) | 65636  (1471) |
| Scene VI | 4924691  (113311) | 352455  (8252) | 172081  (3944) | 1611908  (36433) | 46140  (1090) | 751  (22) | 403970  (9049) |

Table 5 the precision/recall and accuracy of the classification results of scene IV-VI

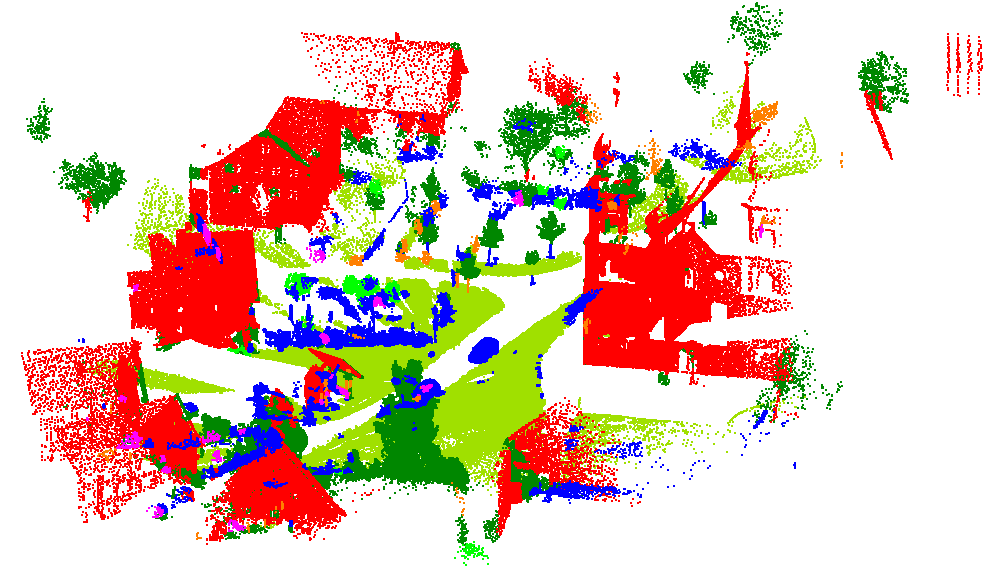
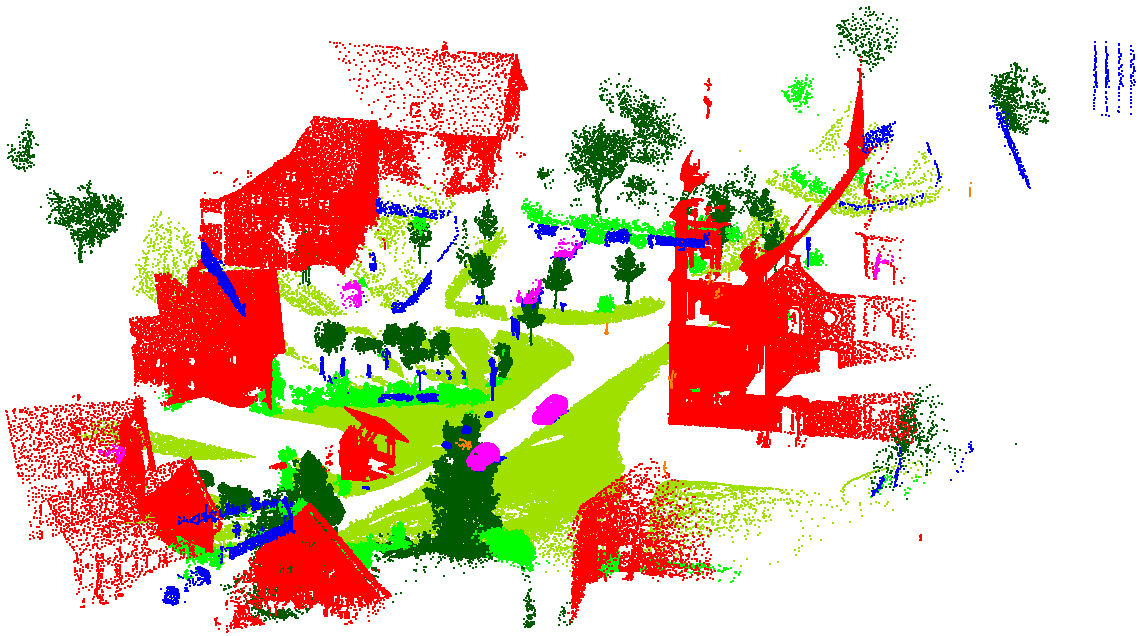
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scene V | natural terrain | high vegetation | low vegetation | buildings | hard scape | scanning artefacts | cars | Accuracy |
| our Method |  |  |  |  |  |  |  |  |
| Method in[17] |  |  |  |  |  |  |  |  |
| Method in[18] |  |  |  |  |  |  |  |  |
| Method in[4] |  |  |  |  |  |  |  |  |
| Method in[35] |  |  |  |  |  |  |  |  |
| Scene VI | natural terrain | high vegetation | low vegetation | buildings | hard scape | scanning artefacts | cars | Accuracy |
| our Method | **98.68/99.52** | **88.14/66.48** | 4.05/31.11 | **92.79/98.46** | **66.61/5.21** | 43.81/2.17 | 1.37/35.13 | **89.31** |
| Method in[17] |  |  |  |  |  |  |  |  |
| Method in[18] |  |  |  |  |  |  |  |  |
| Method in[4] |  |  |  |  |  |  |  |  |
| Method in[35] |  |  |  |  |  |  |  |  |

****

(a)Scene IV (b)training data

****

(c) Scene V (d) result of Scene V

****

(e) Scene VI (f) result of Scene VI

Fig.5 the scene IV-VI, the training data and the results of scene V and VI

The classification results are not as good as the before three scenes. This is because here, the classes are similar for shapes of these classes. Especially for the hard scape, it is an class nearly in concept, as one rock like a car and the real car cannot be easily distinguished by shape. Much worse, there are many hard scape points in scene IV and to fit these points, the DNNSP are overfitting. All the above reasons cause the accuracy in the training process is high but in testing process the accuracy is so low. Many of points are classified to be hard scape and the recall of hard scape is low. Although the car points are not few, there are only three cars which are near the scanner in the scene I, which causes the training data of cars is exactly not enough. Almost all cars are classified to hard scape. The high vegetation and low vegetation are also confused. Half of the low vegetation is classified to hard scape and another half of the low vegetation is classified to high vegetation. However, there is little confusion between the buildings and the vegetation points, which shows although there is little difference between the same type such as the natural terrain, high vegetation and low vegetation but the different types can clearly classification by DNNSP with shape features. Meanwhile, our method relies on the input features, so the features themselves also affect the results. In the future, we will try to use the deep neural network directly extracts features from points.

Compared with the other method, although the accuracy of the classification is not high, our method is still the best and the accuracies improve near 20% comparing with the best of the other methods. This means although the feature are not good for input, the DNNSP can improve the performance of the features for classification.

**C．Performance of Nets in DNNSP**

First, the different number of Net1, Net2 and Net3 are used to shown the effects of the nets. As Net3 is following the Net1 and Net2, to clear show the effects of each net, the Net3 are not used at first and then the results with Net3 are shown. Table 6 shows the results with different nets.

Table 6 the results with different nets（results without net3/with net3）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scene I | | | | | |
| Net1  Net2 | 零个Net1 | 一个Net1 | 两个Net1 | 三个Net1 | 四个Net1 |
| 零个Net2 |  | 96.73+1.54 | 96.98+1.09 | 96.90+0.42 | 96.91+0.58 |
| 一个Net2 | 95.86+2.48 | 96.70+1.59 | **97.42+0.57** | 97.04+0.67 | 97.01+0.43 |
| 两个Net2 | 97.73+0.47 | 96.70+1.11 | 96.72+0.95 | 97.33+0.68 | 96.75+0.71 |
| 三个Net2 | 96.39+1.69 | 97.26+1.06 | 96.68+0.86 | 97.33+0.52 | 96.42+0.70 |
| Scene II | | | | | |
| Net1  Net2 | 零个Net1 | 一个Net1 | 两个Net1 | 三个Net1 | 四个Net1 |
| 零个Net2 |  | 95.05+1.94 | 93.55+1.40 | 93.10+0.89 | 92.91+1.81 |
| 一个Net2 | 85.16+4.38 | **96.53+0.66** | 93.56+1.03 | 94.94+0.72 | 94.53+0.66 |
| 两个Net2 | 80.4+5.26 | 85.18+1.79 | 95.70+0.71 | 92.61+1.81 | 95.27+0.46 |
| 三个Net2 | 85.77+5.56 | 82.11+4.85 | 86.00+2.36 | 88.06+2.07 | 93.61+0.53 |
| Scene III | | | | | |
| Net1  Net2 | 零个Net1 | 一个Net1 | 两个Net1 | 三个Net1 | 四个Net1 |
| 零个Net2 |  | 95.58+0.69 | 93.55+0.97 | 92.55+1.78 | 85.60+3.12 |
| 一个Net2 | 96.64+0.26 | 90.32+1.63 | **96.10+0.36** | 94.71+0.66 | 95.43+0.42 |
| 两个Net2 | 81.56+5.15 | 95.56+0.58 | 87.87+3.44 | 78.75+3.91 | 89.01+2.54 |
| 三个Net2 | 94.04+0.72 | 92.81+0.48 | 94.71+0.61 | 91.56+1.89 | 81.29+2.57 |
| Scene V | | | | | |
| Net1  Net2 | 零个Net1 | 一个Net1 | 两个Net1 | 三个Net1 | 四个Net1 |
| 零个Net2 |  | 85.66+2.16 | 78.43+1.76 | 88.68+1.74 | 87.92.15 |
| 一个Net2 | 85.33+2.53 | 85.53+1.71 | 89.79+2.61 | 89.01+1.46 | **91.02+1.74** |
| 两个Net2 | 80.89+2.38 | 86.56+1.50 | 83.77+2.94 | 87.41+1.52 | 87.46+0.96 |
| 三个Net2 | 82.81+1.59 | 82.21+2.49 | 87.21+2.76 | 89.16+1.84 | 85.31+3.65 |
| SceneVI | | | | | |
| Net1  Net2 | 零个Net1 | 一个Net1 | 两个Net1 | 三个Net1 | 四个Net1 |
| 零个Net2 |  | 86.60+0.44/  86.59+0.56 | 86.57+0.39/  86.50+0.66 | 87.28+0.50/87.21+0.59 | 87.04+0.83/87.89+0.42 |
| 一个Net2 | 84.56+1.32/  84.85+0.55 | 86.68+0.65/  87.52+0.46 | **88.10+0.40/**  87.69+0.37 | 87.53+0.79/87.8+0.33 | 85.92+2.30/88.13+0.63 |
| 两个Net2 | 82.08+2.45/  84.11+0.67 | 84.70+1.79/  **89.27+0.86** | 87.81+0.34/  87.13+0.61 | 86.75+0.68/86.66+1.07 | 86.52+0.87/87.01+0.52 |
| 三个Net2 | 84.83+0.69/  85.39+0.75 | 80.48+1.92/  84.99+1.62 | 83.09+3.30/  87.25+0.64 | 85.09+2.81/87.49+0.94 | 87.66+0.53/  87.24+0.74 |

In the Table, the DNNSP with 0-4 net1 and 0-3 net2 are tested and the best results are highlight by bold type. In summary for the results with and without Net3, there are the best combinations of Net1 and Net2 for the different scenes and after reaching the best result, the results do not improve by the increasing of the number of nets. This means different objects need different number of nets to learn representations and there is no help for so many numbers of nets, if the best results have been reached. Moreover, the accuracies sometimes even decrease, which is maybe because the points are not enough and the overfitting occurs. The best results are all obtained by the combinations of nets, which shows the effects of the Net1 and Net2. For most of the scenes, only one Net2 is enough and the results obtained by two Net1 with one Net2 are the best or near the best. Therefore, two Net1 and one Net2 are used in the paper for point clouds classification. In scene I-III, sometimes the accuracies are suddenly down when the numbers change, which do not occur in scene V and VI. It is maybe because the points are few and sometimes the network convergence to local minimum. It is guessed based the precision of the cars for scene I and II and facades for scene III are near 100% but the recall is low in all these low results.

Compared the results with and without Net3, the result with Net3 are more stable than the results without Net3. Also the best results are improved for most of the scenes. When the number of nets increases, the results with Net3 do not decrease obviously as the results without Net3 and still keep at a high level, which means the Net3 makes sure the results more robust.

To show the effects of extraction of common features in DNNSP, we use the point clusters with the smallest size as the input and use only one full-connect network in Net5. The results are shown in Table 8.

Table 8 the accuracy of classification by using clusters only with one level

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scene I | Scene II | Scene III | Scene V | Scene VI |
| 95.6% | 94.7% | 92.3% | 90.3% | 88.4% |

As we can see the results are obviously lower than use all the levels in scene I-III and the results are close for scenes V and VI. This means the Net5 is helpful for the points are not enough, but for the large points, there is still an increase but slight, which is also similar with the conclusion of the transfer learning. Besides the accuracy of the whole scenes, the precision of scanning artifacts and hard scape in Scene VI decrease to 25% and 54%, when use clusters only in one level. Therefore, for the classes without enough training data, the Net5 are still useful.

.