3D Point Cloud of Single Tree Branches and Leaves Semantic Segmentation Based on Modified PointNet Network

Renpeng Liu\textsuperscript{1}, Lisheng Ren\textsuperscript{1}*, Fang Wang\textsuperscript{1}

\textsuperscript{1}School of Information Science and Technology, Hebei Agricultural University, Baoding, China, 071000

*Corresponding author: rls@hebau.edu.cn

Abstract. Semantic segmentation of single tree 3D point cloud is one of the key technologies in building tree model. It plays an important role in tree skeleton extraction, tree pruning, tree model reconstruction and other fields. Because the area of a single leaf is smaller than that of the whole tree, the segmentation of branches and leaves is a challenging problem. In view of the above problems, this paper first migrates PointNet to the tree branch and leaf point cloud segmentation, and proposes an automatic segmentation method based on improved PointNet. According to the difference of normal direction between leaves and branches, the point cloud information of three dimensions coordinates, color and normal vector is input into the point feature space. In data processing, increase the number of each block data, so that the network can better learn features. MLP is added to the original PointNet network to improve the ability of extracting and learning local features. In addition, in the process of feature extraction, jump connection is added to realize feature reuse and make full use of different levels of features. The original $1 \times 1$ filter of PointNet is replaced by $3 \times 1$ filter to improve the segmentation accuracy of tree point cloud. The focus loss function focal loss is introduced into the field of 3D point cloud to reduce the impact of the imbalance of point cloud samples on the results. The results show that the improved method improves the accuracy of tree branch point cloud segmentation compared with the original PointNet for branch and leaf segmentation. The segmentation accuracy of structural elements of branches and leaves is more than 88%, and MIoU is 48%.

Keywords: Point Cloud, Point Cloud Segmentation, Semantic Segmentation, Deep Learning, PointNet, Tree Branches and Leaves

1. Introduction
In forestry surveying, 3D lidar can accurately and quickly extract the three-dimensional information of the appearance of trees, so it is widely used in the vertical structure of forests, skeleton extraction and three-dimensional tree modeling [1-3]. However, the study of tree branch and leaf segmentation is the premise and important content of tree parameter extraction, tree skeleton extraction and three-dimensional reconstruction. Traditional manual leaf removal has problems such as inefficiency and
inaccuracy. Therefore, it is a key issue that can accurately and quickly segment the branch and leaf point cloud and extract the branch point cloud data by removing the leaves. The traditional branch and leaf point cloud segmentation method has been developed for some time. In traditional branch and leaf point cloud segmentation, there are branch and leaf point cloud segmentation methods based on normal difference operators; branch and leaf point cloud segmentation methods based on fusion of pixel information and point cloud information; based on fusion of multiple spatial structures Feature-based branch and leaf point cloud segmentation method [4-6]. Zhang Yuekun et al. proposed an interactive hierarchical segmentation method based on a height map to extract a single tree point cloud from a sparse tree point cloud [7]. Based on the PC2Tree software, Tang Liyu et al. used a variety of algorithms to separate the branches and leaves of the tree point cloud, and extracted the skeleton structure of the branches and trunks [8]. Li Qicaí and others used super-body clustering to organize and manage the point cloud data structure, and realized single-wood segmentation based on the graph cut algorithm [9]. This method is based on the super-voxel structure to effectively maintain the geometric boundary of the target in the scene and improve the processing efficiency. With the continuous improvement of many researchers, the traditional branch and leaf point cloud segmentation method has been continuously developed. However, the traditional point cloud segmentation method requires manual design of feature descriptors, which requires great professional knowledge of personnel. Traditional branch and leaf point cloud segmentation methods need to select a large number of feature items and threshold items in the segmentation process. The segmentation process is complicated and cumbersome. The time complexity and space complexity of the algorithm are high, and it can only be applied to the segmentation task of specific trees. The generalization ability and robustness are poor. In order to improve the efficiency and automation of point cloud segmentation, the application of deep learning to point cloud segmentation has become a current research trend. Deep learning is a method that uses neural networks to automatically learn and extract features of input data. Since the point cloud is a bunch of scattered and disordered points in space, it is not a regular matrix distribution like an image, so it is very difficult to apply deep learning algorithms on the point cloud [10]. There are four general solutions as follows: ① Convert the point cloud into a multi-view image. Su et al. proposed the MVCNN network, which uses a 2D deep learning framework for 3D object shape recognition. Felix et al. used a multi-data stream full convolutional network to predict the semantic probability of each pixel on the point cloud image under multiple perspectives. ② Voxelizing the point cloud [11]. Point cloud voxelization is to voxelize the disordered point cloud into a regular structure, and then use the neural network structure to perform feature learning to realize the point cloud for semantic segmentation, Daniel et al. proposed a VoxNet network model based on point cloud voxelization [12]. Huang et al. input the pre-processed voxel data into the 3D convolutional neural network to perform voxel-based semantic segmentation. Finally, all points in the voxel will be marked with the same semantic label as the voxel [13]. Although the voxelization method solves the problem of the disorder of the point cloud, this method does have the low efficiency of voxel grid arrangement caused by the sparseness of the point cloud, the large memory occupied in the calculation process, and the long network training time. And the shortcomings such as easy loss of point cloud information. Yang Yuze et al. used the voxel cloud connectivity segmentation algorithm to generate super voxel blocks from the point cloud, and then used the local convex hull connection algorithm to cluster the point cloud based on the concave-convex relationship between different super voxel blocks [14]. So as to realize the division of branches and leaves of trees. ③ Project the point cloud onto a two-dimensional plane. ZHU ZT et al. learn the data features by projecting the point cloud onto the two-dimensional plane, and then input it into a restricted Boltzmann machine stack to extract the features of different projections [15]. The whole network adopts an auto-encoder structure to integrate global features. CAO ZI and others proposed to use spherical projection methods to project around the center of a three-dimensional object and design a network model to use the projected image as the input of a pre-trained convolutional neural network, and use two complementary projection angles to better obtain three-dimensional features [16]. ④ Based on the original point cloud. Stanford University's QI et al. proposed for the first time a PointNet
neural network that can be directly applied to the original point cloud, thereby realizing for the first time that the original point cloud data can be directly input to the network to achieve point cloud classification, partial segmentation and semantic segmentation [17]. Because the network only performs feature extraction for each point, it ignores the important feature of local features, which limits its generalization ability.

This paper uses the improved PointNet to segment the tree branch and leaf point cloud. The main work is as follows: (1) There is currently no public tree point cloud data set. To this end, in order to train the network to better segment the tree branch and leaf point cloud, this paper constructs a tree point cloud data set containing three tree structural units, and provides normal vector information for the experiment. (2) For the first time PointNet network introduces tree point cloud branch and leaf segmentation, and for structural elements with different normal information (leaves, branches), input the coordinate information, color information and normal vector information as tensors into the neural network to improve the segmentation accuracy of the branch and leaf point cloud elements by the PointNet network. (3) Aiming at the problem that the original PointNet network only focuses on extracting the global features of the point cloud and ignoring the local features, the structure of the PointNet network has been improved to make it better to extract local features the ability to improve the segmentation accuracy of tree branches and leaves. (4) Aiming at the problem of the imbalance in the number of point clouds of tree branches and leaves, the focus loss function Focal Loss in the two-dimensional image field is introduced into the three-dimensional point cloud field to replace the conventional cross-entropy loss function to reduce the imbalance of the sample and the segmentation result Impact [18].

2. Construction of tree point cloud data set
Currently, there is no point cloud data set about trees in the public 3D point cloud dataset, and some of the data sets are shown in Table 1. Therefore, in order to better achieve the segmentation of the branch and leaf structure, we created a data set containing 10 trees (North American crabapple trees), 3 semantic elements (leaves, branches, and clutter) and 9, 506, 627 labeled points. The point cloud data set includes not only spatial coordinate information (XYZ), color information (RGB), but also normal vector information (Xn, Yn, Zn) that we have added. For the acquisition of the tree point cloud, we use the TrimbleFX scanner. After obtaining the original target point cloud, perform pre-processing operations such as denoising on the original point cloud, manually label the point cloud according to the format of the S3DIS official data set, and divide the data set into a training set, a verification set and a test set, as shown in Figure 1. Show.

| DataSet     | Year | Format             | Producer              |
|-------------|------|--------------------|-----------------------|
| S3DIS       | 2017 | PointCloud RGB-D   | Stanford              |
| ShapeNet    | 2015 | CAD                | Princeton, Stanford   |
|             |      |                    | and TTIC              |
| Scannet     | 2017 | PointCloud RGB-D   | Stanford              |
| Modelnet40  | 2015 | CAD                | NAVER LABS           |
| vKITTI      | 2016 | PointCloud         |                       |
Figure 1. The main trees of the data set (mainly consists of two parts: the data set with the legal vector and the data set without the legal vector)

3. Method

3.1. PointNet network structure
The PointNet network mainly uses the T-net network to perform rigid transformation, and uses the MaxPool symmetric function to deal with the spatial disorder of the point cloud. The network can be divided into the following three steps: (1) The input data is first aligned by multiplying the transformation matrix learned by a T-Net network to ensure the stability of the model for specific spatial transformations. (2) Use the weight-sharing multi-layer perceptron MLP to extract the features of each point cloud data, then use T-Net to align the features and use the MaxPool symmetry function to pool the disordered point clouds, and finally extract the global features. (3) The local features of each point cloud and the extracted global features are fused in series, and then MLP is used to down-sample the features and calculate the probability score of each point to measure the type of point.

3.2. Improvement of network structure
In view of the fact that the PointNet network is mainly for large scene point clouds, the branch and leaf segmentation effect of tree point clouds is poor, so we improved the original PointNet network. First, we add the normal vector information of the point cloud in the input feature space, and then add a jump connection, then calculate the neighborhood of each point in the point cloud, and add a layer of MLP to extract local features, and then fusion of global features and local features. The input point cloud contains N points of dimension D. Each point in the point cloud is input into the MLP, and each layer of the MLP network is batch normalized, and ReLU is used as the activation function. The MLP network can extract the C-dimensional feature of each point, and after the maximum pooling operation, obtain the C-dimensional global feature of the point cloud. The revised network structure is shown in Figure 2.
3.2.1. Preprocessing of input data
This paper uses Trimble FX laser scanner to scan the tree point cloud, and uses its matching TrimbleRealWork software to perform processing operations such as denoising and registration. After that, the processed point cloud is manually labeled with tree trunks and leaves, and it is made into a tree point cloud data set in the official format of S3DIS.

3.2.2. Feature space input
Due to the sparse structure of the point cloud and the disorder of the point cloud, we need to convert the point cloud into a format that the neural network can understand. Three-dimensional coordinates (X, Y, Z) represent spatial information, and colors (R, G, B) represent color information. Normalize the coordinates of the original point cloud before inputting. The point cloud is divided according to its type, and then normalized according to the coordinates of the trees. After the coordinates (X0, Y0, Z0) are obtained, each tree is divided into small blocks of 10cm×10cm, and then each small block is predicted the semantic label of each point. In addition, unlike the S3DIS data set input, we have added the normal vector information of the trees, because the branches and leaves contain different normal information. Finally, each point in the point cloud is input (X, Y, Z, R, G, B, X0, Y0, Z0, Xn, Yn, Zn) 12-dimensional feature space to the neural network.

3.2.3. Extract local features
In the process of optimizing the original PointNet network, we found that adding multiple layers of MLP to the original PointNet network helps to improve the accuracy of branch and leaf segmentation. Therefore, we added a 2-layer MLP to the feature network of PointNet to extract a single point, which together with the original extraction layer serves as the extraction layer for local features. In addition, a jump connection is added behind the original MLP-4 to fuse the original feature value with the extracted feature to achieve feature reuse, so as to make full use of features at different levels. Compared to the original Pointnet, improved network structure uses more MLP layers and a more complex structure to extract local features, which improves the ability to extract and learn local features. When optimizing the MLP layer to extract high-dimensional features, we find that a larger convolution kernel will obtain better accuracy. Therefore, we replace the 1×1 convolution kernel part of the original PointNet with a 3×1 convolution kernel to improve the segmentation accuracy of the tree point cloud. In order to better segment leaves, we added neighborhoods. We use the KNN nearest neighbor algorithm to find the local neighborhood of each point first, then extract local features from each neighborhood, and then fuse them with global features. Here we use Euclidean distance to calculate the distance between points. The Euclidean distance calculation formula is shown in (1), and we use the Concat function of the tensorflow framework to achieve the fusion of neighborhood features and global features. The algorithm of Concat function is shown in (2). Compared with the original PointNet's feature extraction of a single point, this method makes full use of the neighboring features of the point.

K-nearest neighbors: For any sampling point Q, there are k points adjacent to it and the nearest distance. According to the distance from Q to the nearest to farth order, the permutation α is satisfied, then the neighborhood point mathematics of K-nearest neighbors. The expression is the longest distance in the neighborhood, and the K nearest neighborhood is shown in Figure 3 [19].
3.2.4. Design of loss function

In the network model, the loss function is an indicator that measures the difference between the predicted value and the true value. In many deep learning task data sets, there are often samples of certain categories that are much larger than samples of other categories, and the resulting sample imbalance will affect the effect of model learning. In the point cloud data, due to the influence of the leaves and branches in the scanner and the actual trees, different degrees of sample imbalance will also be caused. For example, the data volume of leaves in the data set scenario in this article is much more than the data volume of branches and other elements. In order to solve the problem, this paper introduces the focus loss function (Focal Loss) in the two-dimensional image field into the three-dimensional point cloud field to replace the conventional cross entropy loss function (Cross Entropy Loss) to reduce the impact of sample imbalance on the final result. The definition formula of Focal Loss is shown in (3):

\[ J = -\alpha_i (1 - p_i)^\gamma \log(p_i). \]  

(3)

Among them, \( \alpha \) represents the weight factor, \( p_i \) represents the predicted probability of the network input through the softmax calculation process, \( (1 - p_i)^\gamma \) represents the modulation coefficient, and \( \gamma \) represents the hyperparameter in the modulation coefficient, where \( \gamma \geq 0 \).

4. Result

We use the unmodified PointNet network and the improved PointNet network to semantically segment the tree point cloud data set we created, and use the k-fold cross validation method to select the tree point cloud data from Tree1 to Tree10 as the test set, and other data as the train set. Training set to verify the effectiveness of our improved network. We compared the segmentation results using the improved PointNet and the original PointNet segmentation results. We also compared the results of using normal vectors and no normal vectors in the data set. The deep learning framework we use is Tensorflow, the optimizer is adam to train the network, the learning rate is set to 0.001, the batch-size is set to 8, and the epoch is set to 50. The equipment we used to train the network is shown in Table 2 and Table 3.
| Hardware | CPU | GPU | RAM |
|----------|-----|-----|-----|
|          | Intel(R)Xeon(R) E5-1620 | Nvidia GTX1060(4G) | 16G |

| Software | Operating system | Deep learning framework | GPU accelerator | Programme language |
|----------|------------------|-------------------------|-----------------|--------------------|
|          | Window 7         | Tensorflow1.1.0          | CUDA 8.0        | Python 3.5         |
|          |                  |                         | CuDNN 5.1       |                    |

### Table 2. Experimental platform hardware

### Table 3. Experimental platform software

#### 4.1. Test results

The tree point cloud data we constructed is shown in Figure 4. Use the original PointNet network to segment the tree data set we constructed, and the result is shown in Figure 5. The result of the improved PointNet network segmentation of the tree dataset we constructed is shown in Figure 6. Figure 4 shows the input tree point cloud data, and Figures 5 and 6 are the results of segmentation. We compare the single tree results of the original network and the improved network (Figure 7), and the results show that the improved PointNet network is better than the original PointNet in segmenting the tree branch and leaf point cloud.

![Figure 4. Original tree point clouds (partially)](image)

![Figure 5. Segmentation result of original network](image)

![Figure 6. Segmentation result of improved network](image)

![Figure 7. Comparison of single tree results](image)
Figure 5. PointNet split results on our data set (partially)

Figure 6. The results of the segmentation of the improved network on our data set (partially)

Figure 7. Comparison of the original network with the improved network single tree results

4.2. Performance

Compared with point cloud classification and recognition, 3D point cloud semantic segmentation needs to finely identify the semantic category of each point, which is a very challenging task. In order to further evaluate the ability of the network model in this paper to handle fine-grained tasks of three-dimensional point clouds, this paper uses the commonly used semantic segmentation evaluation indicators Intersection Ratio (IOU), Average Intersection Ratio (MIOU) and Overall Accuracy (OA) to measure segmentation. The results of, they are respectively defined by formulas (4)–(6):

\[
IOU = \frac{T_P}{T_P + F_P + F_N}.
\]  

(4)
\[
MIOU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{T_p}{T_p + F_p + F_N}
\]

\[
OA = \sum_{i=0}^{k} \frac{T_p'}{N_p'}
\]

Among them, \( k \) it represents the number of divided categories, \( T_p \) represents the number of positive samples that are classified correctly, \( F_p \) represents the number of positive samples that are incorrectly classified, \( F_N \) represents the number of negative samples that are incorrectly classified, \( T_p' \) represents the correct number of divisions for the class, and \( N_p' \) represents the total number of points in the point cloud. \( N_p' \) represents the total number of points in the point cloud of the class.

4.3. Result analysis
Table 4 compares the results of the improved network and the original PointNet segmentation of the dataset with normal vectors and non-vectors. In our data set, we use the improved PointNet algorithm and the original PointNet algorithm to segment the OA of each category and the overall MIOU. We found that adding normal vector information to the point cloud, whether using the improved network or the original PointNet, helps to improve the segmentation accuracy of the branch and other segmentation. For the improved network, except for leaf, the accuracy and the average intersection ratio are overall higher than that of PointNet. Table 5 shows the results of our improved network using different loss functions. The results show that the accuracy and MIOU obtained by using the focal loss function are greatly improved compared with the cross-entropy loss function. Table 6 shows that increasing the PointNet part of the convolution kernel can improve the overall segmentation effect on the tree branch and leaf segmentation. Therefore, the method and improvement in this article are effective.

**Table 4. PointNet and the improved network split accuracy on our data set**

| Network                  | OA(%) | MIOU(%) | leaf | branch |
|--------------------------|-------|---------|------|--------|
| PointNet(without Normal) | 85.61 | 47.16   | 84.52| 37.44  |
|                          |       |         | 15.43|        |
| PointNet(with Normal)    | 88.32 | 46.81   | 88.32| 36.68  |
|                          |       |         | 19.51|        |
| Our(with Normal)         | 88.57 | 48.41   | 87.91| 41.83  |
|                          |       |         | 15.46|        |
| Our(without Normal)      | 88.06 | 40.67   | 87.57| 32.43  |
|                          |       |         | 12.01|        |

**Table 5. The improved network uses different loss functions to split the results on Tree6**

| Loss Function       | OA(%) | MIOU(%) |
|---------------------|-------|---------|
| Cross Entropy Loss  | 86.56 | 37.43   |
| Focal Loss          | 90.54 | 45.14   |

**Table 6. The improved network uses neighbourhood segmentation results on our data set**

| Neighbourhood       | OA(%) | MIOU(%) |
|---------------------|-------|---------|
| Our(without neighbourhood) | 85.16 | 47.12   |
5. Conclusion

Aiming at the low efficiency of manual segmentation of branch and leaf point clouds and the poor ability of PointNet network to extract local features of tree branch and leaf point clouds, an improved PointNet-based end-to-end deep neural network is proposed for semantic segmentation of tree 3D laser point clouds. According to the structure of the PointNet network, we replaced the original 1×1 filter with a 3×1 filter, and added a multi-layer MLP, and added the neighborhood of each point when extracting features to improve local feature extraction capabilities. In addition, a jump connection is added. By connecting features of different layers, low-level features and high-level features are merged to realize feature reuse, and make full use of features at different levels. Aiming at the problem of unbalanced data category samples, the focus loss function of the two-dimensional image domain for the unbalanced data samples is introduced into the point cloud domain to replace the conventional cross-entropy loss function. Finally, we compare the improved PointNet and the results obtained by using the original PointNet on the tree data set we constructed. The results show that, compared with the original PointNet, the improved PointNet has better accuracy and higher MIoU, which proves that the improvement is effective.

Acknowledgements

This research was financially supported by the Key Research and Development Project of Agricultural Intelligent Equipment of Hebei Key Laboratory of Agricultural Big Data (Grant NO. 19227120D) and the Major technological innovation project for the revitalization of dairy cows (Grant NO.19220119D).

References

[1] Wang Qi, Hu Hong, Wu Yanlan, etc. Automatic extraction method of tree diameter at breast height based on 3D laser point cloud [J]. Journal of Anhui Agricultural University, 2017, 44(2): 283-288.
[2] Lin Zhidong, He Yuanrong, Zhuang Laoguo. A single tree skeleton point extraction method based on SX10 point cloud [J]. Bulletin of Surveying and Mapping, 2020(03):160-162.
[3] Xie D H, Wang X Y, Qi J B, et al. Reconstruction of single tree with leaves based on terrestrial LiDAR point cloud data [J]. Remote sensing, 2018, 10(5): 1-18.
[4] Li S H, DAI L Y, Wang H S, et al. Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model [J]. Remote sensing, 2017, 9(12): 1-16.
[5] Wang Hongshu, Li Shihua, Guo Jiawei, et al. Inversion of leaf area density of magnolia forest canopy based on ground-based lidar [J]. Journal of Remote Sensing, 2016, 20(4): 570-578.
[6] Yu Yaoshen, Yun Ting, and Yang Xubing. Multi-dimensional feature segmentation method of plant organs based on laser point cloud data [J]. Data Collection and Processing, 2015, 30(5): 1054-1061.
[7] Zhang Yuekun, Yu Wenjie, Zhao Xizhi, Lv Yanxing, Feng Weihuan, Li Zhengrong, Hu Shaojun. Research on Interactive Tree Segmentation and Modeling Method Based on Airborne Lidar Point Cloud [J/OL]. Journal of Graphics: 1 -10 [2021-03-03].
[8] Tang Liyu, Zhang Hao, Huang Hongyu and others. Improved tree 3D reconstruction method based on point cloud data [J]. Journal of Agricultural Machinery, 2017, 48(2): 186-194.
[9] Li Qicai, Zhao Chung's surname. Laser point cloud tree segmentation combined with supervoxel and graph optimization [J]. Science of Surveying and Mapping, 2020, 45(09): 117-122+147.
[10] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3D shape recognition [C]// 2015 IEEE International Conference on Computer Vision (ICCV),
December 7-13, 2015, Santiago, Chile, New York: IEEE, 2015: 945-953.

[11] LAWIN F J, DANELLIAN M, TOSTEBER G P, et al. Deep projective 3D semantic segmentation [C]/International Conference on Computer Analysis.

[12] Daniel M and Sebastian S., 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, Congress Center Hamburg, Sept 28-Oct 2. Hamburg, Germany, 922-928.

[13] HUANG J, YOU S. Point cloud labeling using 3D convolutional neural network [C]/23rd International Conference on Pattern Recognition, Cancun, 2016: 2670-2675.

[14] Yang Yuze, Lin Wenshu. Tree branch and leaf segmentation and 3D reconstruction based on laser point cloud data [J]. Journal of Northwest Forestry University, 2020, 35(03): 171-176.

[15] ZHU Z T, WANG X G, BAI S, et al. Deep learning representation using autoencoder for 3D shape retrieval [J]. Neurocomputing, 2016, 204: 41-50.

[16] CAO Z J, HUANG Q X, KARTHIK R. 3D object classification via spherical projections [C]/2017 International Conference on 3D Vision, Qingdao, 2017: 566-574.

[17] Charles R. Qi, Hao Su, Kaichun Mo, etc. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [R]. Stanford: Stanford University, 2016.

[18] LIN Tsung-Yi, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [C]/Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy: IEEE Computer Society Press, 2017: 2980-2988.

[19] Guskov I, Sweldens W, Schroder P. Multiresolution signal processing for meshes [A]. In Computer Graphics Proceedings (SIGGRAPH 99) [C]. 1999, 325-334.