Segmentation of Plant Point Cloud based on Deep Learning Method

Yibin Lai[1], Shenglian Lu[1], Tingting Qian [2], Ming Chen[1], Song Zhen[1], and Li Guo[1]

[1]Guangxi Key Lab of Multisource Information Mining & Security, School of Computer Science and Technology, Guangxi Normal University, Guilin 541004, China, laiyibee@163.com
[2]Institute of Agricultural Science and Technology Information , Shanghai Academy of Agriculture Sciences, Shanghai 201403, China, qiantingting@saas.sh

Corresponding author: Shenglian Lu, shllu@126.com

Abstract. The acquisition of plant phenotypic parameters from three-dimensional(3D) point cloud often requires a certain amount of manual intervention and manual setting of thresholds, which brings limitations to the automatic high-throughput plant phenotyping. This paper introduces a deep learning-based method for segmenting plant parts from 3D point cloud. The training network used in our approach is a point cloud instance segmentation network based on 3D bounding box regression called 3D-BoNet. This network consists of a backbone and two branch network: 1)The backbone network is used for extracting the local feature and global feature of point cloud. 2)Two branch networks include the bounding box prediction network and the point-level mask prediction network. The instance bounding box of point cloud is predicted through the bounding box prediction network by inputting point cloud feature into the association layer and multi-criteria loss function, and the instance point-level mask is predicted through the point-level mask prediction network. Finally, our method combines the network, point cloud reconstruction method, data annotation, and data augmentation to complete specific segmentation task and the final segmentation results are evaluated. The experimental results show that the augmentation dataset based on rotation is effective for leaf segmentation, and the improvement of the backbone network also shows a significant effect on the instance segmentation network. For the individual leaf segmentation task, the best mean accuracy is 0.809, and the mean recall rate is 0.884; while for the individual plant segmentation task, the best mean accuracy is 0.875, the mean recall rate is 0.897.

Keywords: Deep learning, Point cloud, Plant phenotyping, 3D-BoNet, Instance segmentation, Semantic segmentation
DOI: https://doi.org/10.14733/cadaps.2022.1117-1129

1 INTRODUCTION

Automatic measuring and monitoring technologies make a great revolution in 21st-century. High-throughput, precise, and non-destructive plant phenotype measurement has become a key issue in the agricultural field. It is usually difficult to obtain plant phenotype information through images to fully reflect the true spatial morphology of crops. Therefore, researchers seek to obtain 3D structure information of plants from point clouds [7, 8]. Point cloud is a kind of data that can represent 3D depth information. Many point cloud-based applications emerged not only in the plant phenotyping but also in many other fields such as autonomous driving [3], robot perception and grabbing, and 3D map reconstruction. Compared with two-dimensional(2D) image which has an unreal scale and easy to be affected by space occlusion, three-dimensional(3D) point cloud will bring object recognition many advantages. In recent years, deep learning-based methods for object detection and recognition from image or video have achieved amazing results [9]. However, due to the irregularity and disorder of point cloud data, the application of deep learning on 3D point cloud faces many limitations and challenges, and there is still potential for exploration.

The existing deep learning-based methods for point cloud data [4]mainly developed from two aspects. One is processing point cloud through indirect or mapping methods, such as multi-view method [12] and voxel-based method [18]. The multi-view method maps the point cloud to multi-view images and then uses image convolution for classification. The voxel-based method rasters point cloud into voxels first, and then the feature of per voxel is extracted using 3D convolution method. The other is a point-based method, a classic deep learning network for point cloud called PointNet [11], which provides an end-to-end learning method that can directly process point cloud. The network is simple and scalable. It is useful, but it cannot capture the local feature of point cloud, so PointNet++ [10] proposes iterative feature extraction through combining the field information of each point so that the network can extract the local feature of the point cloud better.

In plant phenotyping research, we want to automatically obtain plant phenotypic parameters through three-dimensional point clouds. One of the key steps is to segment the point clouds of different scenes by instance. We assume that the point cloud instance of each plant in the scene can be obtained by an instance segmentation, then we can obtain the number, position, and size of the plants through previous instance segmentation results. SGPN [13] is the first to use a neural network algorithm to group point-level embeddings on the 3D point cloud for instance segmentation. And ASIS [14] also used the same strategy to reorganize point cloud features to achieve instance segmentation. However, the objects segmented by these methods do not have good objectivity, and the boundaries are not detected clearly. And the methods such as GSPN [17] and 3D-SIS [6] implement 3D instance segmentation methods based on proposal-based. However, they usually need two-stage training and post-processing methods such as Region Proposal Network(RPN) [15] and Non-Maximum Suppression(NMS) [5] to prune dense proposals. We notice that the point cloud instance segmentation network called 3D-BoNet [16]. 3D-BoNet is a point cloud instance segmentation network based on directly acting on three-dimensional point cloud data. This network can predict the point-level mask of each instance while regressing the bounding box of each instance in the point cloud so that it will predict the result of point cloud instance segmentation. This network structure is simple and efficient, has good scalability, and does not require post-processing methods of general target detection network such as Region Proposal Network(RPN) [15] or Non-Maximum Suppression(NMS) [5], it directly processes the point cloud data without projecting to voxel grids or images, and the rate of processing the point cloud reaches a point cloud with about 4000 points per 20ms. The above characteristics of the network are worth researching on the application of plant phenotypic parameters acquisition based on deep learning with this network. The workflow of this paper is shown in Fig.1.

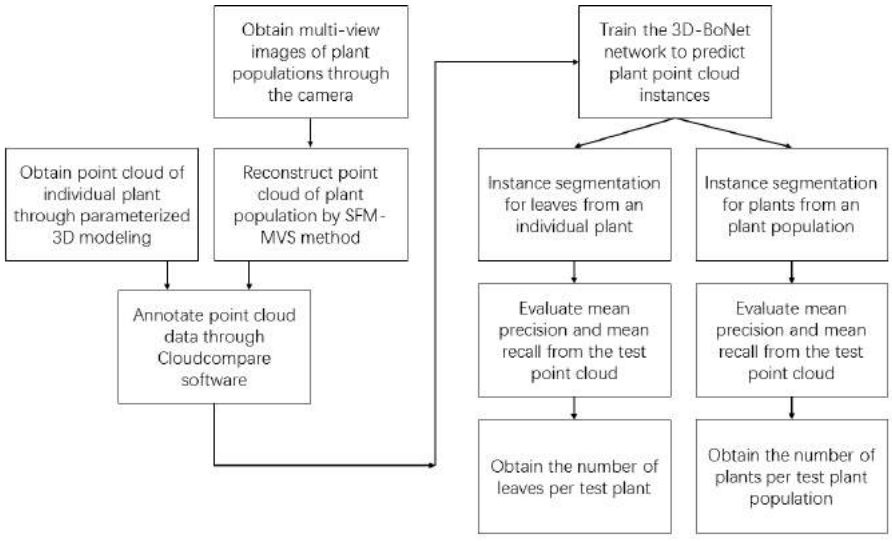


Figure 1: The workflow of this paper.

2 PLANT DATASETS COLLECTION AND ANNOTATION

2.1 Plant Images Collection and Point Cloud Reconstruction

Three crops were chosen for collecting images, including cucumbers, eggplants, and tomatoes. All the images of these crops were collected from indoor and outdoor environments. For the scene of plant population, each crop was placed according to 3 rows * 3 columns and 4 rows * 4 columns under different intervals (30, 40, 50, 60 cm) for multi-view images collection. The multi-view images of each plant population or individual were shot from different positions in a hemispherical shape around the target plant. As shown in Fig.2, the vertical and horizontal separation distance between positions of the adjacent camera on the hemispherical surface is approximately $20 \sim 30$ °. In addition, a few extra images were also taken from the top so that these images can cover the entire target plant.

We used the SFM-MVS (structure from motion and multi-view stereo) method to reconstruct the 3D point cloud. The collected images were input into VisualSFM [2] software to automatically reconstruct the 3D point cloud of plant population, the processing as shown in Fig.3. And the point cloud of different plants under different periods are shown in Fig.4 and Fig.5.

Besides the point clouds of plant population generated from multi-view images, a group of point clouds of individual cucumber plant is also prepared. These point clouds are obtained through parameterized 3D modeling, as shown in the Fig.6.

2.2 Segmentation Tasks for two Situations

In this study, we consider two segmentation tasks from the 3D point cloud:

(1) Segmenting plant individual from point cloud of a plant population: This scene restricts our inputting the original point cloud to a plant population that contains the ground. The scene of plant population is displayed like a 9-square grid. We have tried to place different plants, plant numbers, and adjust plant spacing to obtain point clouds data of different plant populations. To meet the need of monitoring the growth of plants, it is necessary for us to automatically determine how many plants exist in this scene.

(a) Original cucumber photo.

(b) Original eggplant photo.

(c) Original tomato photo.

Figure 2: The original photos were taken using a Canon EOS 5D Mark III digital camera.

From a semantic perspective, we need to do a binary classification task, that is, to separate the ground from the scene; from an instance perspective, we not only separate the ground from the scene but also separate different individual plants in this plant population. In this way, we can count the number of individual plants and the point cloud of each individual can also be used for further organ segmentation.

(2) Segmenting organs from point cloud of an individual plant: This application is to segment each organ (e.x. leaf, fruit) from the point cloud of an individual plant, this segmentation would further be processed for extracting some phenotypic parameters, such as leaf length, leaf width, the number of each category organ, and so on.

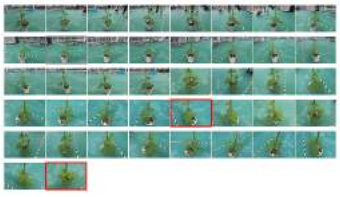
2.3 Semantic Annotation and Instance Annotation of Plant Point Cloud

After obtaining the point clouds of these plant population scenes through the SFM-MVS method, we performed the annotation work on these point clouds similar to the S3DIS dataset [1] to obtain data that can be trained and tested. For each point in the point cloud, we have the information of x, y, z, r, g, and b. We use the segment tool on Cloudcompare software to divide the points of different semantic labels and different labels of instance into different parts, as shown in Fig.6 and Fig.7. While using the 3D-BoNet network, we also need to convert these point clouds into corresponding h5 files to meet the requirements of the input data.

3 POINT CLOUD INSTANCE SEGMENTATION NETWORK

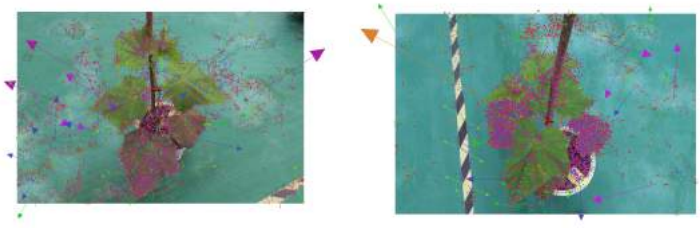
3.1 The Main Workflow of 3D-BoNet Network

The main structure of 3D-BoNet is divided into a backbone network and two branch networks:

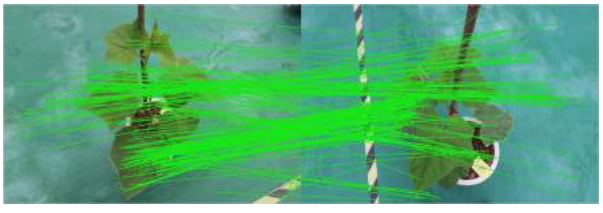


(a) The 42 multi-view photos of the same plant.

(b) Two photos of the same cucumber.



(c) The Sift feature from the photo.



(d) The match process with SIFT feature.

(e) Sparse point cloud generated by SFM method.

(f) Dense point cloud generated by MVS method.

Figure 3: The process of SFM-MVS method to reconstruct plant point cloud.

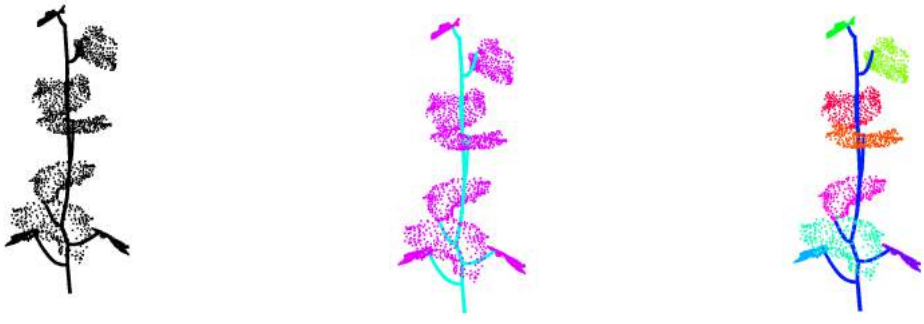
- **Backbone network:** The backbone network is mainly responsible for the feature extraction of point clouds. In 3D-BoNet, PointNet or PointNet++ is used as the backbone network to extract the local

(a) Original cucumber point cloud. (b) Original eggplant point cloud. (c) Original tomato point cloud.

Figure 4: Point cloud of different kinds of plants.

(a) Acquired on April 13 (b) Acquired on April 21. (c) Acquired on April 27. (d) Acquired on May 12.

Figure 5: Point clouds of cucumber plants acquired in different periods.

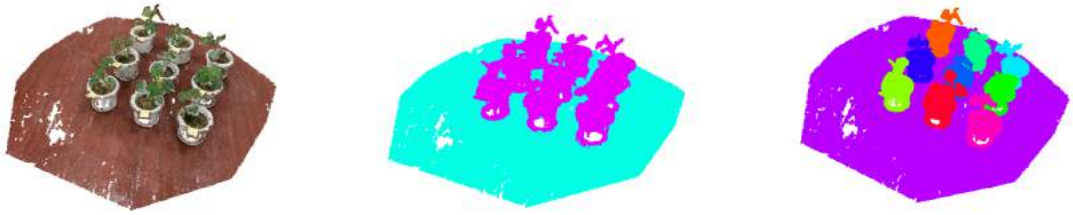


(a) Individual cucumber plant point cloud. (b) Annoatation with semantic label. (c) Annoatation with instance label.

Figure 6: Annoatation of single plant point cloud.

and global features of the point cloud for input to the subsequent network branch.

- **The bounding box prediction branch:** The core part of 3D-BoNet is the bounding box prediction branch. The main purpose of this branch is to predict the bounding box of the instance through the global feature extracted from the backbone network, without the need for pre-defined spatial anchors or the use of Region Proposal Network(RPN). To predict and learn the bounding box of this instance, it is necessary to establish a connection between the predicted bounding box and the real labeled box.



(a) Plant population point cloud. (b) Annoatation with semantic label. (c) Annoatation with instance label.

Figure 7: Annoatation of plant population point cloud.

In this branch, the connection is established by proposing an association layer and then connecting a multi-criteria loss function. This simultaneous process allows this branch to predict the bounding box of each instance independently and undirectionally.

– **Association layer:** The association layer is to establish a connection between the predicted bounding box and the ground truth bounding box. In this layer, we need to pay attention to who is and who is related, and what is the cost if they are associated. The framework proposes two matrices to describe this problem. $\boldsymbol{A}$ is a boolean incidence matrix, and $\boldsymbol{C}$ is an incidence cost matrix. The association problem of the bounding box is finding the allocation matrix $\boldsymbol{A}$ with the smallest total cost, which is formulated as follows:

$$\boldsymbol{A} = \underset{\boldsymbol{A}}{argmin} \sum_{i=1}^{H}\sum_{j=1}^{T} \boldsymbol{C}_{i,j}\boldsymbol{A}_{i,j} \quad subject \quad to \quad \sum_{i=1}^{H} \boldsymbol{A}_{i,j} = 1, \sum_{j=1}^{T} \boldsymbol{A}_{i,j} \leq 1, j \in \{1..T\}, i \in \{1..H\} \tag{1}$$

– **Multi-criteria loss function:** In the association cost matrix, the association cost between the predicted bounding boxes and the ground truth bounding boxes needs to be calculated. In other words, the association cost indicates the degree of similarity between the two boxes. The more similar the two boxes, the lower the association cost. To calculate this similarity, three criteria are proposed in the framework:

1) The similarity of Euclidean distance:

$$\boldsymbol{C}_{i,j}^{ed} = \frac{1}{6}\sum(\boldsymbol{B}_i - \overline{\boldsymbol{B}}_j)^2 \tag{2}$$

2) Soft IoU:

$$\boldsymbol{C}_{i,j}^{sIoU} = \frac{-\sum_{n=1}^{N}(q_i^n * \overline{q}_i^n)}{\sum_{n=1}^{N} q_i^n + \sum_{n=1}^{N} \overline{q}_i^n) - \sum_{n=1}^{N}(q_i^n * \overline{q}_i^n)} \tag{3}$$

3) Cross entropy score:

$$\boldsymbol{C}_{i,j}^{ces} = -\frac{1}{N}\sum_{n=1}^{N}[\overline{q}_i^n \log q_i^n + (1-\overline{q}_i^n)\log(1-q_i^n)] \tag{4}$$

In general, criteria (1) guarantees the boundary learned by the prediction box, and criteria (2)(3) can maximize the coverage of effective points and overcome non-uniformity. The sum formula of the three criteria as follow:

$$\boldsymbol{C}_{i,j} = \boldsymbol{C}_{i,j}^{ed} + \boldsymbol{C}_{i,j}^{sIoU} + \boldsymbol{C}_{i,j}^{ces} \tag{5}$$

Formulation of loss function as follows:

$$l_{bbox} = \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{C}_{i,j}^{ed} + \boldsymbol{C}_{i,j}^{sIoU} + \boldsymbol{C}_{i,j}^{ces}) \tag{6}$$

– **Bounding box score prediction:**

$$l_{bbs} = -\frac{1}{H}\left[\sum_{t=1}^{T}\log \boldsymbol{B}_s^t + \sum_{t=T+1}^{H}\log(1-\boldsymbol{B}_s^t)\right] \tag{7}$$

- **Prediction point mask branch:** The global features and local features were obtained in the backbone network, as well as the instance bounding box feature was obtained in the prediction bounding box branch, these features finally were fed to the prediction point mask branch. The purpose of this branch is to determine the points in the prediction box is it a valid point or a background point.

3.2 Experiments

We train the network of instance and semantic segmentation on the plant datasets based on this network. The semantic segmentation branch is the same as used in [13]. While using the training set of different scenes, we changed the two parts in the training process.

- **Training environment and parameter settings:** Our training and testing are running on GTX 1660 super graphics card and Intel i5 CPU core under Ubuntu16.04, tensorflow1.4, python3.6, and Cuda 8.0. The settings is consistent with the original 3D-BoNet network. There is a H hyperparameter that represents the maximum number of predicted instance bounding boxes in a scene and we set it to 20. The initial default learning rate is 0.0005. As the epoch increases by 20, the learning rate will be divided by 2. The settings of the training datasets and testing datasets are shown in the Tab.1.
- **Using different backbone networks:** The network of 3D-BoNet does not limit the use of any kind of backbone network. The idea in the point-based method dominated by PointNet has been widely used in subsequent point cloud networks. However, the original PointNet network cannot capture the local characteristics of point clouds, so the subsequence PointNet++ is taken into account the local feature of each point's neighbors. In our training experiment, we use different backbone networks to evaluate the difference.
- **Data rotation augmentation based on point cloud:** The previous PointNet work has the characteristics of taking into account the invariance of point cloud rotation. Before the point cloud is input to the network for training, a T-transform network is used to adapt to the effect of point cloud rotation invariance. This method is not completely effective. In the follow-up PointNet++, the T-transform network was also discarded. The leaves in our plant point cloud dataset have obvious changes in the rotation characteristics, and the angles and directions of each leaf are varied. Usually, it is more difficult to solve the invariance of point cloud rotation characteristics only by the network itself. Therefore, we need to augment the point cloud rotation of the dataset. After rotation augmentation in different directions and different angles, these differences are used to train the network. We rotate the point cloud around the X-axis, Y-axis, Z-axis, or X-Y-Z axis to create new data. These new data and the original point cloud are put into the same folder as the new dataset. The number changes of point cloud after rotation augmentation as shown in the Tab.2.

	Number of training point clouds	Number of testing point clouds	Maximum number of point clouds	Minimum number of point clouds
Single plant dataset(number)	32	4	51422	29384
Population plant dataset(number)	16	4	1800287	407367

Table 1: Training datasets and testing datasets settings

	Number of original point clouds	After X-axis rotation augmentation	After X-axis+Y-axis+Z-axis+XYZ-axis rotation augmentation	After XYZ-axis rotation augmentation
Single plant dataset(number)	32	64	128	\
Population plant dataset(number)	16	\	\	32

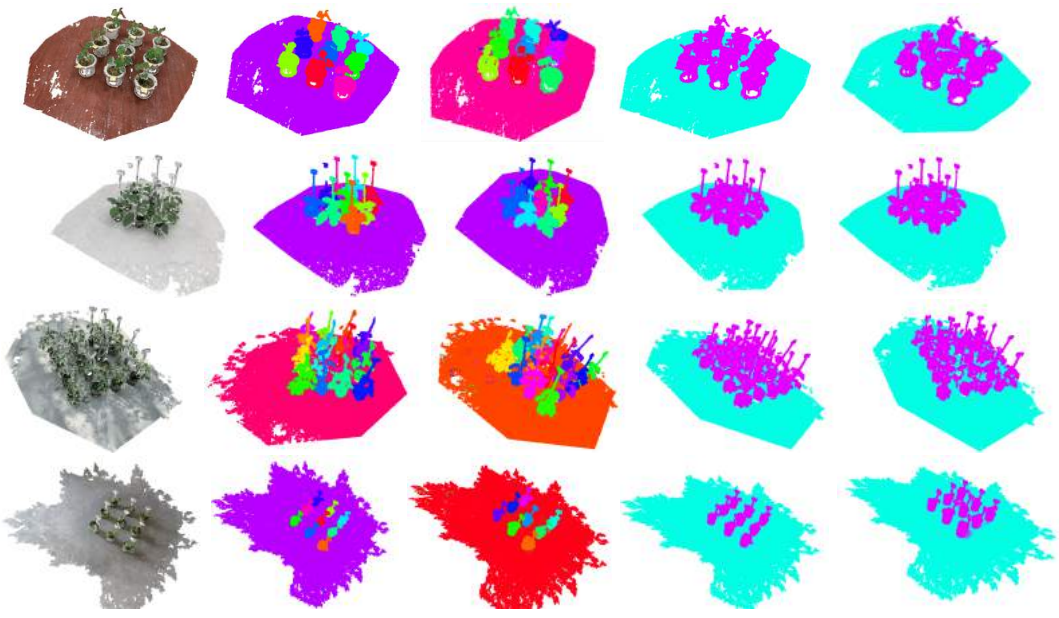
Table 2: Number changes of training point cloud after rotation augmentation.

4 RESULTS AND EVALUATIONS

After 100 epochs of training each dataset had completed, the obtained network model was used to predict semantic and instance labels on the test point cloud. The visual results of the best prediction showed in the Fig.8 and Fig.9; while the threshold of IoU was set to 0.5 to evaluate mean precision(mPrec) and mean recall(mRec), we got the results shown in Tab.3 and Tab.4. Whether in the segmentation of plant population or individual plant, the effect of semantic prediction showed good consistency. Compared with the indoor spaces dataset(S3DIS) [1], the dataset of plant has fewer categories such as whole plant, leaf organs, ground, and branches. The results of the instance prediction showed that: (1)in the scene of the plant population, although most of them are in line with our expectation, it is still the case where two plants are predicted to be the same plant(in the Fig.8 1 row and 3 columns); (2)in the scene of the individual plant, there is the case where the edge part between two leaves is predicted to be the same leaf(as shown in the Fig.9 2 rows and 3 columns).

	original dataset		dataset with XYZ-Axis rotation augmentation	
backbone	mPrec	mRec	mPrec	mRec
PointNet	0.788	0.667	\	\
PointNet++	**0.875**	**0.897**	0.829	0.743

Table 3: The mean precision and mean recall under different backbone networks and different augmentation plant population datasets.



(a) original point cloud (b)ins_label (c)ins_pred (d) sem_label (e)sem_pred

Figure 8: Segmentation results of plant population point cloud.

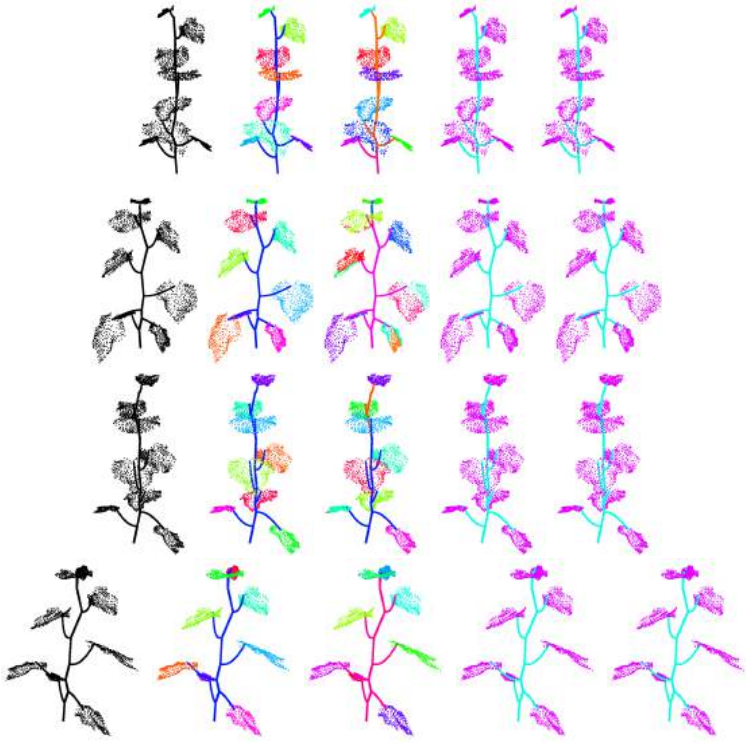
	original dataset		dataset with X-Axis rotation augmentation		dataset with X-axis+Y-axis+Z-axis+XYZ-axis rotation augmentation	
backbone	mPrec	mRec	mPrec	mRec	mPrec	mRec
PointNet	0.543	0.442	0.571	0.55	0.674	0.721
PointNet++	0.636	0.488	0.634	0.634	**0.809**	**0.884**

Table 4: The mean precision and mean recall under different backbone networks and different augmentation single plant datasets.

The acquisition of plant phenotypes is what we need to pay attention to. For example, the number of plants and leaves in the scene can be obtained by counting the number of instances. Therefore, the number of instances in the prediction results is compared with the number of ground truth instances. The comparison results are shown in Fig.10.

5 CONCLUSIONS

According to the above results and evaluation, we can reach the following conclusions: (1) From the vertical direction of the Tab.3 and Tab.4, more detailed point cloud features can bring benefits to instance segmentation. This is in line with the expected law. Therefore, the research on improving the performance of the point cloud extraction network is still a dynamic research direction. Most segmentation networks need to perform feature extraction on the point cloud firstly. (2) For point cloud data augmentation based on rotation, it can be roughly concluded from the horizontal direction of Tab.3 and Tab.4: the angle of leaf distribution in an



(a) original point cloud (b)ins_label (c)ins_pred (d) sem_label (e)sem_pred

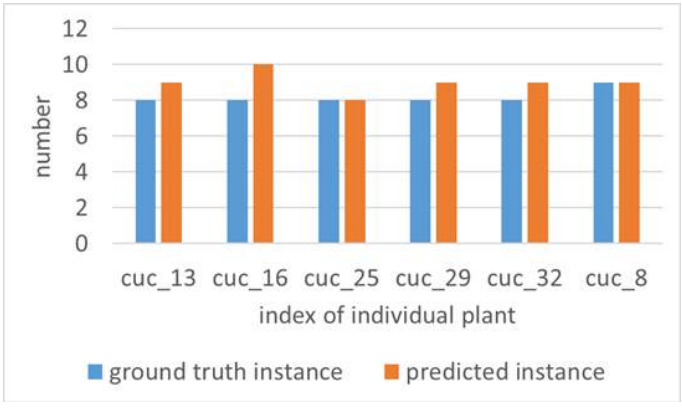
Figure 9: Segmentation results of single plant point cloud.

individual plant changes greatly, more effective expansion can be obtained after data augmentation based on rotation. On the plant population, the relative position of the ground and the plants is relatively fixed. Since there are not many data sets, the data augmentation based on rotation will confuse the relationship between the ground and the plants, thereby reducing the segmentation effect.

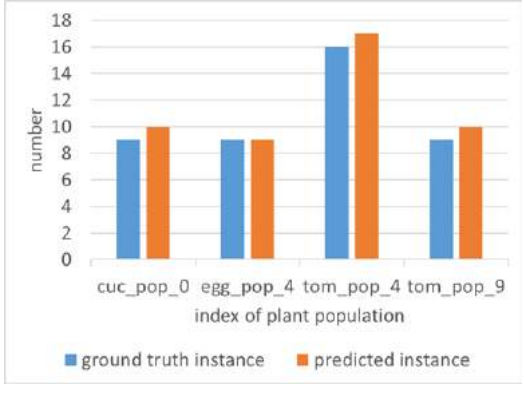
This paper explores the application of deep learning in plant point cloud instances and semantic segmentation, which provides direction for the subsequent research on the automatic acquisition of plant phenotypic parameters. We found that in the related work of the point cloud network, the data used for training is critical and valuable, and often requires a large number of samples, and the cost of obtaining and labeling these data is too high. The related research of semi-supervised learning based on small samples may contribute to this direction. Plant phenotype parameters such as the number of pots or leaves, the plant height information of each plant, and the leaf length and leaf width contained in each leaf, the acquisition of above parameter require accurate segmentation of each instance, therefore it brings many challenges. The potential of deep learning-based point cloud processing in plant phenotyping research maintains a hot topic and open challenge, which deserves more people's attention.

ACKNOWLEDGEMENTS

This research is supported by National Natural Science Foundation of China (No. 61762013, 62062015 and 61662006), the Science and Technology Foundation of Guangxi Province (No. 2018AD19339) and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (No. 20-A-02-02)



(a) Prediction results of leaf instances from a plant.



(b) Prediction results of plant instances from a population.

Figure 10: Comparison of the number of ground truth instances and the predicted instances.

Yibin Lai, http://orcid.org/0000-0002-2485-5957
Shenglian Lu, http://orcid.org/0000-0002-4957-9418
Tingting Qian, http://orcid.org/0000-0001-9708-2521
Ming Chen, http://orcid.org/0000-0003-0506-5308
Song Zhen, http://orcid.org/0000-0002-5197-3772
Li Guo, http://orcid.org/0000-0001-5539-7824

REFERENCES

[1] Armeni, I.; Sener, O.; Zamir, A.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S.; IEEE: 3d semantic parsing of large-scale indoor spaces, 2016. http://doi.org/10.1109/CVPR.2016.170.

[2] Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R.: Towards internet-scale multi-view stereo, 2010. http://doi.org/10.1109/CVPR.2010.5539802.

[3] Geiger, A.; Lenz, P.; Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. 3354–3361, 2012. http://doi.org/10.1109/CVPR.2012.6248074.

[4] Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Bennamoun, M.: Deep learning for 3d point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(99), 1–1, 2020.

[5] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.: Mask r-cnn. International Conference on Computer Vision, 2017.

[6] Hou, J.; Dai, A.; Niener, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. Computer Vision and Pattern Recognition (CVPR), 2018.

[7] Hui, F.; Zhu, J.; Hu, P.; Meng, L.; Zhu, B.; Guo, Y.; Li, B.; Ma, Y.: Image-based dynamic quantification and high-accuracy 3d evaluation of canopy structure of plant populations. Annals of Botany, 5(121), 1079–1088, 2018. http://doi.org/10.1093/aob/mcy016.

[8] Jin; Shichao; Yanjun; Fangfang; Pang; Shuxin; Gao; Shang; Tianyu; Liu: Stem-leaf segmentation and phenotypic trait extraction of individual maize using terrestrial lidar data. IEEE Transactions on Geoscience and Remote Sensing, 3(57), 2019. http://doi.org/10.1109/TGRS.2018.2866056.

[9] Krizhevsky, A.; Sutskever, I.; Hinton, G.: Imagenet classification with deep convolutional neural networks. 1, 1097–1105, 2012.

[10] Qi, C.R.; Li, Y.; Hao, S.; Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Neural Information Processing Systems, 2017.

[11] Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. Computer Vision and Pattern Recognition, 77–85, 2017. http://doi.org/10.1109/CVPR.2017.16.

[12] Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. International Conference on Computer Vision, 945–953, 2015. http://doi.org/10.1109/ICCV.2015.114.

[13] Wang, W.; Yu, R.; Huang, Q.; Neumann, U.: Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. Computer Vision and Pattern Recognition, 2017. http://doi.org/10.1109/CVPR.2018.00272.

[14] Wang, X.; Liu, S.; Shen, X.; Shen, C.; Jia, J.: Associatively segmenting instances and semantics in point clouds, 2019. http://doi.org/10.1109/CVPR.2019.00422.

[15] Wolter, S.; Schüttpelz, M.; Tscherepanow, M.; Linde, S.; Heilemann, M.; Sauer, M.: Real-time computation of subdiffraction-resolution fluorescence images. Journal of Microscopy, 237(1), 12–22, 2010. http://doi.org/10.1111/j.1365-2818.2009.03287.x.

[16] Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S.; Markham, A.; Trigoni, N.: Learning object bounding boxes for 3d instance segmentation on point clouds. Neural Information Processing Systems, 2019.

[17] Yi, L.; Zhao, W.; Wang, H.; Sung, M.; Guibas, L.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. Computer Vision and Pattern Recognition (CVPR), 2018.

[18] Yin, Z.; Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. Computer Vision and Pattern Recognition, 2017.