AAAS
Plant Phenomics
Volume 2022, Article ID 9787643, 20 pages
https://doi.org/10.34133/2022/9787643

Research Article

PSegNet: Simultaneous Semantic and Instance Segmentation for Point Clouds of Plants

Dawei Li,[1,2] **Jinsheng Li**,[3] **Shiyu Xiang**,[3] **and Anqi Pan**[2,3]

[1]*State Key Laboratory for Modification of Chemical Fibers and Polymer Materials, College of Information Sciences and Technology, Donghua University, Shanghai 201620, China*
[2]*Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, China*
[3]*College of Information Sciences and Technology, Donghua University, Shanghai 201620, China*

Correspondence should be addressed to Anqi Pan; pananqi@dhu.edu.cn

Received 1 January 2022; Accepted 7 April 2022; Published 23 May 2022

Phenotyping of plant growth improves the understanding of complex genetic traits and eventually expedites the development of modern breeding and intelligent agriculture. In phenotyping, segmentation of 3D point clouds of plant organs such as leaves and stems contributes to automatic growth monitoring and reflects the extent of stress received by the plant. In this work, we first proposed the Voxelized Farthest Point Sampling (VFPS), a novel point cloud downsampling strategy, to prepare our plant dataset for training of deep neural networks. Then, a deep learning network—PSegNet, was specially designed for segmenting point clouds of several species of plants. The effectiveness of PSegNet originates from three new modules including the Double-Neighborhood Feature Extraction Block (DNFEB), the Double-Granularity Feature Fusion Module (DGFFM), and the Attention Module (AM). After training on the plant dataset prepared with VFPS, the network can simultaneously realize the semantic segmentation and the leaf instance segmentation for three plant species. Comparing to several mainstream networks such as PointNet++, ASIS, SGPN, and PlantNet, the PSegNet obtained the best segmentation results quantitatively and qualitatively. In semantic segmentation, PSegNet achieved 95.23%, 93.85%, 94.52%, and 89.90% for the mean Prec, Rec, F1, and IoU, respectively. In instance segmentation, PSegNet achieved 88.13%, 79.28%, 83.35%, and 89.54% for the mPrec, mRec, mCov, and mWCov, respectively.

1. Introduction

Plant phenotyping is an emerging science that connects genetics with plant physiology, ecology, and agriculture [1]. It studies a set of indicators formed by the dynamic interaction between genes and the growth environment to intuitively reflect the growth of plants [2]. The main purpose of the research is to accurately analyze the relationship between phenotypes and genotypes by means of computerized digitization, to improve the understanding of complex genetic traits, and eventually expedite the development of modern breeding and precision agriculture [3–5]. Generally speaking, the analysis of plant phenotypes mainly focuses on organs, including the aspects such as leaf characteristics, stem characteristics, fruit traits, and root morphology. As the organ with the largest surface area, leaves serve for the main place of photosynthesis and respiration [6]. Therefore, the leaf area, leaf length, width, and the leaf inclination are among the most critical phenotypic factors [7]. In addition to leaves, the stem system not only forms the skeleton of the plant structure but also spatially connects all other organs such as leaves, flowers, and fruits. The phenotyping of the stems can reflect the extent of stress received by the plant.

The key to plant phenotyping is to segment plant organs efficiently and correctly. Since the 1990s, a flow of researches have emerged upon the task of plant organ segmentation, especially the leaf segmentation for disease recognition. The 2D image-based phenotyping is usually based on traditional image processing, machine learning, and pattern recognition algorithms, such as the threshold-based segmentation [8, 9], edge detection [10, 11], region growing [12, 13], clustering [14, 15], and their combinations and extensions [16–20]. In recent years, deep learning based on

convolutional neural networks (CNNs) has reached state of the art on image classification and image segmentation [21–23]. References [24–29] applied image deep learning networks to segment fruits and leaves from plant images. However, the 2D phenotyping methods usually deal with simple rosette plants (e.g., *Arabidopsis* or tobacco) or several monocotyledonous plants with fewer leaves (e.g., wheat and maize). The main reason is that a 2D image is taken from only one viewing angle, missing the depth information. And the occlusion and overlapping between leaves in the canopy bring huge challenges to the segmentation algorithms based on 2D images. Moreover, images cannot fully describe the complete spatial distribution of the plant structure, resulting in a less reliable statistical significance of the measured phenotypic traits.

Compared with images, 3D models not only contain the information of color and texture but can also carry the most important information—the depth. Depth directly overcomes the problems caused by occlusion and overlapping, which are becoming the basis for high-precision phenotypic measurement. In recent years, with the development of low-cost and high-precision 3D imaging technology, plant phenotyping methods based on depth images or point clouds are quickly emerging. As the 3D imaging technique with the highest precision, Light Detection and Ranging (Lidar) is now widely used for 3D reconstruction and phenotyping of tall trees [30, 31], maize [32, 33], cotton [34], and several other cash crops [35–37]. 3D sensors based on Structured Light and Time-of-Flight (ToF) have also become two important means for 3D phenotyping on plants due to their remarkable real-time performances [38, 39]. References [40, 41] reconstructed and analyzed a variety of greenhouse crops and cash crops using binocular stereo vision. References [42, 43] carried out 3D reconstruction and phenotypic analysis on crops by using the multiview stereo (MVS) technique. Miao et al. designed a toolkit—Label3DMaize [44], for annotating 3D point cloud data of maize shoots; the toolkit facilitates the preparation of manually labeled maize 3D data for training and testing on machine learning models.

Despite the precise 3D data, how to effectively separate plant individuals from the cultivating block and how to further divide each plant into organs to calculate phenotypic parameters are two difficult tasks in phenotyping. Unsupervised leaf segmentation on 3D point clouds has already begun to attract interests. Paproki et al. [45] improved the point cloud mesh segmentation algorithm and proposed a hybrid segmentation model that could adapt to the morphological differences among different individuals of cotton, and they achieved the separation of leaves from stems. Duan et al. [46] used the octree algorithm to divide the plant point cloud into small parts and then manually merged each part into a single organ according to their spatial topological relationships. Itakura and Hosoi [47] utilized the projection method and the attribute extension for leaf segmentation; they also tested the segmentation accuracy of seedlings of 6 types of plants. Su et al. [48] and Li *et al.* [49] used the Difference of Normal (DoN) [50] operator to segment leaves in point clouds of magnolia and maize, respectively. In addition, Zermas et al. [51] proposed to use node information in the 3D skeleton of plant to segment the overlapping maize leaves. The abovementioned point cloud segmentation and phenotyping techniques still lack generality on segmentation of different crop species with diverse leaf shapes and canopy structures. Meanwhile, their applications are sometimes restricted by the complicated parameter tuning in segmentation. Pan et al. [52] and Chebrolu *et al.* [53] used spatiotemporal matching to associate the organs in growth for phenotypic growth tracking.

Designing a general 3D segmentation method for multiple plant species at different growth stages is the current frontier of 3D plant phenotyping. With the recent breakthrough in artificial intelligence, deep learning-based segmentation methods for unorganized and uneven point clouds are becoming popular across both academics and the agricultural industry. Previous studies mainly focused on the multiview CNNs [54–58] that understand the 3D data by strengthening the connection between 2D and 3D by CNNs on images. However, two issues exist in multiview CNNs, i.e., it is hard to determine the angle and quantity of projection from a point cloud to a 2D image, and the reprojection from the segmented 2D shapes back to the 3D space is not easy. Some studies resorted to a generalization from 2D CNNs to the voxel-based 3D CNNs [59–63]. In 3D CNN, the point cloud is first divided into a large number of voxels and 3D convolutions are used to achieve direct segmentation on the point cloud. However, the computational expense of this method is high. PointNet [64] and PointNet++ [65] operate directly on points and are able to simultaneously conduct classification and semantic segmentation at point-level. Since then, improvements on the PointNet-like framework were made to enhance the network performance mainly by optimizing and/or redesigning the feature extraction modules [66]. Masuda [67] applied PointNet++ to the semantic segmentation of tomato plants in greenhouse and further estimated the leaf area index. Li et al. [68] designed a PointNet-like network to conduct semantic and instance segmentation on maize 3D data. Similarity group proposal network (SGPN) [69] devised a similarity matrix and group proposals to realize simultaneous instance segmentation and semantic segmentation of point clouds. Graph neural networks (GNNs) [70–74] obtained information between adjacent nodes by converting the point cloud into a connective graph or a polygon mesh.

So far, deep learning has becoming a promising solution for high-precision organ segmentation and phenotypic trait analysis of plant point clouds. However, several problems are yet to be solved—(i) the lack of a standardized downsampling strategy for point clouds that are specially prepared for deep learning; (ii) the network design for multifunctional point cloud segmentation is challenging—e.g., a network is hard to keep balance between the organ semantic segmentation task and the instance segmentation task; and (iii) the lack of generalization ability among different species; e.g., a good segmentation network for monocotyledonous plants may not work properly on dicotyledonous plants.

To address the above challenges, a deep learning network—PSegNet, was designed to simultaneously conduct

plant organ semantic segmentation and leaf instance segmentation on a manually labeled point cloud dataset of multiple species. PSegNet obtained state-of-the-art results on two kinds of dicotyledonous plants (tobacco and tomato) and a monocotyledonous plant—sorghum. The detailed contributions are stated as follows:

(i) We proposed Voxelized Farthest Point Sampling (VFPS), a novel point cloud downsampling strategy that possesses advantages from both Voxelization-based Sampling (VBS) and Farthest Point Sampling (FPS). VFPS is suitable to be used to prepare diversified dataset for training of deep neural networks because it can easily augment point cloud data by the random initialization in sampling. Ablation experiments "A6" showed that the proposed VFPS strategy significantly improved the accuracies of organ semantic segmentation and instance segmentation for several varieties of crops by contrasting with the traditional FPS

(ii) A deep learning network—PSegNet, was specially designed for segmenting point clouds of several species of plants. After training on the dataset prepared with VFPS, the network can simultaneously realize the semantic segmentation of the stem class and the leaf class and the instance segmentation for each single leaf. Comparing to several mainstream deep learning networks such as PointNet++ [65], ASIS [75], SGPN [69], and PlantNet [76], our PSegNet obtained the best segmentation results qualitatively and quantitatively. The effectiveness of the modules in the architecture of PSegNet, including the Double-Neighborhood Feature Extraction Block (DNFEB), the Double-Granularity Feature Fusion Module (DGFFM), and the Attention Module (AM), was verified separately by the ablation study

The notations and nomenclatures used in this paper are summarized in Table 1. The rest of the paper is arranged as follows. Materials and related methods are explained in Section 2. Comparative experiments and results are given in Section 3. Some further discussions and analysis are provided in Section 4. The conclusion is drawn in the last section.

Table 1: Notations and nomenclatures.

FPS	Farthest Point Sampling
VBS	Voxelization-based Sampling
VFPS	Voxelized Farthest Point Sampling
DNFEB	Double-Neighborhood Feature Extraction Block
DGFFM	Double-Granularity Feature Fusion Module
AM	Attention Module
CA	Channel attention
SA	Spatial attention
DHL	Double-hinge Loss
GT	Ground truth
MLP	Multilayer perceptron
ReLU	Rectified linear unit activation
PE	Position encoding
EC	EdgeConv operation
AP	Attentive pooling
F_c, F_f	Feature maps after decoding
F_{DGF}	Aggregated feature map after DGFFM
$L, L_{\mathrm{sem}}, L_{\mathrm{ins}}, L_{\mathrm{DHL}}, L_s, L_d, L_{\mathrm{reg}}$	The loss functions
C	The number of semantic classes
N	The number of points in a point cloud
$\mathbf{p}_i$	A point in XYZ space
$\mathbf{f}_i, \mathbf{r}_i, \mathbf{h}_i, \mathbf{f}'_i, \mathbf{fe}_i$	A point vector in feature space
K	The parameter of KNN
$\alpha, \beta, \gamma, \delta_s, \delta_d$	Parameters for loss functions
$\cup$	Feature concatenation
max [•]	The maximum value across the inputs
IoU[•, •]	IoU of the two entities
MLP[•]	MLP operation with shared parameters

2. Materials and Methods

2.1. Point Cloud Sampling. There are usually tens of thousands of points in a high-precision point cloud, and the processing of such complicated point clouds will inevitably incur a huge computation burden. Moreover, the majority of modern neural networks for 3D learning only accept standardized input point clouds, i.e., a fixed number of points for all point clouds. Therefore, it is necessary to carry out sampling before substantial processing and modeling. There are two commonly used methods for point cloud downsampling—the Farthest Point Sampling (FPS) [65] and the Voxel-based Sampling (VBS) [77]. FPS first takes a point from the original point cloud P to form a point set A, and each time then traverses all points of the set $P \setminus A$ to find the farthest point p from A. Finally, the point p is taken out from $P \setminus A$ into A, and the iteration ends till the point set A has reached the number limit. This method can maintain the local density of the point cloud after sampling with a moderate computational cost. The disadvantage of FPS is that it may easily lose details of areas that are already sparse and small. VBS constructs voxels in the 3D space of the point cloud, and the length, width, and height of the voxel are defined as three voxel parameters l_x, l_y, l_z, respectively. VBS uses the center of gravity of each voxel to replace all original points in that voxel to achieve downsampling. Despite the high processing speed, there are two main disadvantages of VBS—(i) the three parameters for the scale of voxelization need to be adjusted according to the density and the size of the point cloud, making the number of points

after VBS sampling to be uncertain. Therefore, VBS cannot be directly adopted on the batch processing of a large point cloud dataset. And (ii) VBS creates evenly distributed point clouds, which is unfavorable for the training of deep neural networks whose performances rely on the diversity of the data distribution.

After studying the strengths and weaknesses of FPS and VBS, a new point cloud downsampling strategy—Voxelized Farthest Point Sampling (VFPS), is proposed. The VFPS strategy can be divided into three steps shown as Figure 1. The first step is to determine N, the number of downsampled points. The second step is to adjust the voxel parameters to conduct VBS on the original point cloud to generate a point cloud having slightly more points than N. The last step is to apply FPS on this temporary voxelized point cloud, and downsample is a result with N points. In all experiments of this paper, we fix N at 4096. VFPS possesses advantages from both VBS and FPS; it can not only generate standardized points but can also easily generate diversified samples from the same original point cloud by starting FPS from a randomly chosen point, which is highly suitable to be used as a data augmentation measure for the training of deep learning networks.

2.2. Network Architecture. The overall architecture of PSegNet is shown in Figure 2. The end-to-end network is mainly composed of three parts. The front part has a typical encoder-like structure that is frequently observed in deep neural networks, and the front part contains four consecutive Double-Neighborhood Feature Extraction Blocks (DNFEBs) for feature calculation. Before each DNFEB, the feature space is downsampled to reduce the number of points and to condense the features. The design of the middle part of the network is inspired from Deep FusionNet [78]. We name this part as Double-Granularity Feature Fusion Module (DGFFM), which first executes two parallel decoders, in which the upper decoder is a coarse-grained decoder, while the lower decoder has more layers, representing a fine-grained process. At the end of DGFFM, concatenation operation and convolutions are carried out to realize feature fusion. The third part of the network has a double-flow structure. The feature flow of the upper branch and the feature flow of the lower branch correspond to the instance segmentation task and semantic segmentation task, respectively. Features on each flow of the third part pass through the Attention Module (AM) that contains a spatial attention and a channel attention operation, so that the two flows can aggregate desirable semantic information and instance information, respectively. In the output part of PSegNet, the semantic flow obtains the predicted semantic label of each point through the argmax operation on the last feature layer to complete the semantic segmentation task. The instance flow performs MeanShift clustering [79] on the last feature layer to realize the instance segmentation.

2.3. Double-Neighborhood Feature Extraction Block. DNFEB was designed to improve the feature extraction in the front part of PSegNet. The front part contains 4 consecutive DNFEBs. And in this encoder-like front part, DNFEB follows immediately after a downsampling of the feature space each time. DNFEB has two characteristics: (i) it pays attention to the extraction of both high-level and low-level features at the same time; (ii) by continuously aggregating the neighborhood information in the feature space, the encoder can realize comprehensive information extraction from the local scale to the global one. The detailed workflow of DNFEB is shown in Figure 3, which contains two types of K-nearest neighborhood calculations and three similar stages in a row.

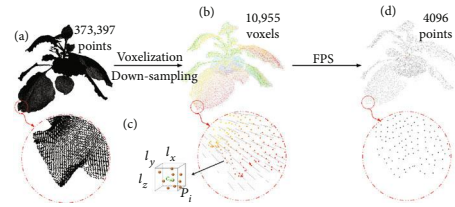

Figure 1: Schematic diagram of the VFPS strategy. The leftmost point cloud (a) is an original tobacco point cloud that contains a total of 373,397 points. First, we set a number object for downsampled point cloud, e.g., $N = 4096$. Then, the VBS with parameters $l_x = l_y = l_z = 3$cm is applied to the original point cloud to form a point cloud (b) containing 10955 voxels. Each voxel is represented by the center of gravity of points in the voxel, and a voxel example is enlarged in (c). At last, FPS is applied on this temporary voxelized point cloud to generate the final result (d) with an exact 4096 points.

The lower part of Figure 3 enlarges the detailed calculation steps of the stage 1 of DNFEB. Take the i-th point as example, its original XYZ space coordinate (the low-level feature space) is represented by $\mathbf{p}_i$, and the feature vector of its current feature space is $\mathbf{f}_i$. The K-nearest neighbors searched in the initial XYZ space are defined as a set $\{\mathbf{p}_{i1}, \cdots, \mathbf{p}_{iK}\}$, while the K-nearest neighbors in the current feature space are denoted by $\{\mathbf{f}_{i1}, \cdots, \mathbf{f}_{iK}\}$. The low-level neighborhood features are calculated by the Position Encoding (PE) operation to obtain the encoded low-level neighborhood feature set $\{\mathbf{r}_{i1}, \cdots, \mathbf{r}_{iK}\}$, in which the feature vector is calculated by (1):

$$\mathbf{r}_{iK} = \mathbf{p}_i \bigcup \mathbf{p}_{ik} \bigcup (\mathbf{p}_i - \mathbf{p}_{ik}) \bigcup \|\mathbf{p}_i - \mathbf{p}_{ik}\|_2. \quad (1)$$

In Equation (1), the operator $\cup$ represents vector concatenation, and $\|\cdot\|_2$ represents the L2-norm. Therefore, $\mathbf{r}_{iK}$ is a 10-dimensional low-level vector, representing the position information of the k-th point near $\mathbf{p}_i$. The neighborhood of the current feature space is calculated by the EdgeConv (EC) [74] operation to obtain a high-level neighborhood feature set $\{\mathbf{h}_{i1}, \cdots, \mathbf{h}_{ik}\}$. The calculation method of $\mathbf{h}_{ik}$ in the set is given in (2).

$$\mathbf{h}_{ik} = \mathrm{MLP}\left[\mathbf{f}_i \bigcup (\mathbf{f}_i - \mathbf{f}_{ik})\right], \quad (2)$$

where MLP[·] represents a multilayer perception operation with shared parameters.

The encoded low-level neighborhood $\{\mathbf{r}_{i1}, \cdots, \mathbf{r}_{ik}\}$ is concatenated with the high-level neighborhood $\{\mathbf{h}_{i1}, \cdots, \mathbf{h}_{ik}\}$,

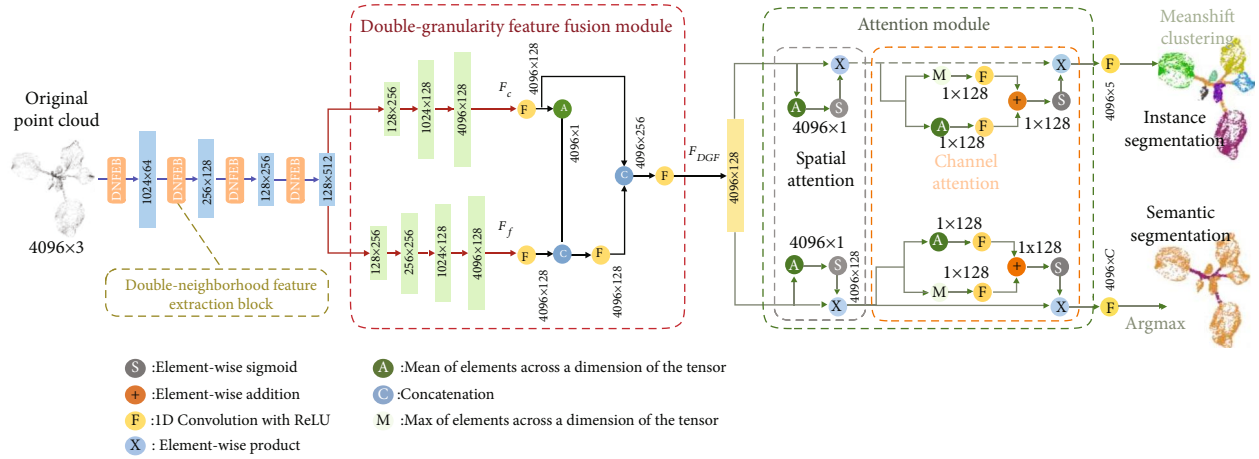

FIGURE 2: The architecture of PSegNet. The network is mainly composed of three parts. The front part has a typical encoder-like structure in deep learning. Four consecutive Double-Neighborhood Feature Extraction Blocks are applied in the front part computation, and the feature space is downsampled before each DNFEB to condense the features, respectively. The middle part is the Double-Granularity Feature Fusion Module, which fuses the outputs of two decoders with different feature granularity to obtain the mixed feature F_{DGF}. In the third part of PSegNet, the features flow into two directions that, respectively, correspond to two tasks—instance segmentation and semantic segmentation. Spatial attention and channel attention mechanisms are sequentially applied on each feature flow.

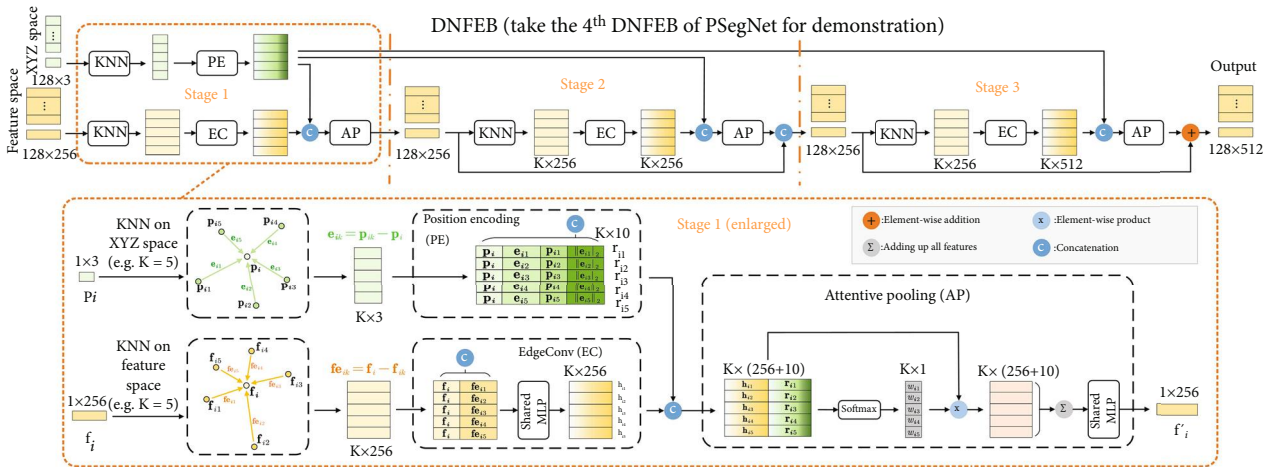

FIGURE 3: Demonstration of DNFEB. The feature dimensions in DNFEB vary with their positions in the PSegNet, and in this figure, we only display feature dimensions of the 4th DNFEB. A standard DNFEB contains three similar stages. The calculation process of stage 1 is enlarged in the lower part of the figure. On stage 1, for any point i in the feature space, we find its K-nearest neighbors in the initial XYZ space and in the current feature space, respectively. Secondly, position encoding is carried out for K-nearest neighbors in XYZ space to form a low-level feature encoding of the local region. At the same time, EdgeConv is carried out for the K-nearest neighbors in the current feature space to form a high-level feature representation of the local region. Finally, after concatenating the low-level and high-level local features, the new feature vector of the current point i is output after the calculation of the Attentive Pooling operation.

and the combination then aggregates into a single feature vector $\mathbf{f}_i'$ by the Attentive Pooling (AP) operation. The calculation process can be represented by the following equation.

$$\mathbf{f}_i' = \mathrm{MLP}\left[\sum_{k=1}^{K} w_{ik} \cdot \left(\mathbf{h}_{ik} \bigcup \mathbf{r}_{ik}\right)\right], \tag{3}$$

in which the weight set $\{w_{ik} | k = 1, \cdots K\}$ is obtained by Softmaxing on the concatenated features of the low-level and high-level feature sets. The following addition of all features in the neighborhood realizes information aggregation. Equation (3) can be regarded as a generalized form of average pooling. Due to the introduction of attention mechanism, its effect is better than ordinary average pooling, which has become a standard operation in deep learning. After the calculation of the AP module, the vector $\mathbf{f}_i'$ becomes the output of the i^{th} point on stage 1. After all points have completed the AP process on stage 1, the set $\{\mathbf{f}_1', \cdots, \mathbf{f}_N'\}$ will become the input of stage 2 of DNFEB to continue the calculation. Though the three stages seem similar, it should be noted that several tiny differences exist among them. For example, the

output of stage 1 is concatenated with the output of stage 2; however, the output of stage 2 is directly added with stage 3 in order to reduce the amount of parameters. And, there is also no skip connection on stage 1.

2.4. Double-Granularity Feature Fusion Module. Deep FusionNet [78] believes that both the coarse-grained voxel-level features and the fine-grained point-level features are of great significance to feature learning. This network strengthens its feature extraction ability after fusing the features under two different granularities. Inspired by Deep FusionNet, we design a Double-Granularity Feature Fusion Module (DGFFM) in the middle part of our PSegNet. In DGFFM (shown in Figure 2), two parallel decoders are created to construct feature layers with two different learning granularities, in which the upper decoder with less layers simulates the coarse-grained learning process, while the lower decoder with more layers simulates the fine-grained learning. After obtaining the coarse-grained feature map F_c and the fine-grained feature map F_f, the module combines them through operations such as average pooling and feature concatenation and finally obtains the aggregated feature F_{DGF} after 1D convolution with ReLU. The output of DGFFM can help to improve the performances of the semantic and instance segmentation tasks that follow.

2.5. Attention Module and Output Processing. Attention has become a promising research direction in deep learning field, and although it has been widely applied to 2D image processing [80–84], the use of attention mechanism on 3D point cloud learning networks is still in its infancy. Inspired by [81], we design an Attention Module (AM) that contains two subattention mechanisms, i.e., the spatial attention (SA) and the channel attention (CA). Spatial attention (SA) tends to highlight points that can better express meaningful features among all input points and gives higher weights to those more important feature vectors for strengthening the role of key points in the point cloud representation. Channel attention (CA) tends to focus on several dimensions (channels) of the feature vector that encode more interesting information. The importance of each feature channel is automatically judged via pooling along the feature channel direction, and then, each channel is given a different weight according to its importance to strengthen the important feature dimensions and meanwhile suppress the unimportant dimensions.

In the structure of AM, the feature calculation is divided into two parallel ways, the feature flow of the upper branch performs the instance segmentation task, while the feature flow of the lower branch corresponds to the semantic segmentation task. Both feature flows first pass the calculation of SA and then the CA. Although the calculation measures are the same for the two flows, the parameters of parallel convolutions are not shared. In SA, we create a vector of 4096×1 by carrying out average pooling on the output feature 4096×128 from DGFFM and then conduct sigmoid function on the vector to obtain the spatial weight vector. By multiplying the weight vector with the DGFFM feature, the feature strengthened by SA is obtained, and its size is 4096×128. In the following CA, we obtain two vectors with size 1×128 by averaging pooling and maxing pooling along the channel direction of the SA output, respectively. And then, two vectors are added together after 1D convolutions. Finally, we use sigmoid function to obtain a 1×128 channel weight vector. By multiplying the weight vector with the pooled feature vector, we obtain the strengthened feature vector of the SA mechanism, which is also the output of a flow of the AM module.

For the point cloud instance segmentation task, the instance flow outputs a 4096×5 feature map after AM calculation with an extra 1D convolution. Then, the MeanShift clustering algorithm [79] is used on the feature map to generate the instance segmentation result. During training, the loss of instance segmentation is calculated immediately after clustering. For the semantic segmentation task, by using an extra 1D convolution and an Argmax operation on the semantic flow output feature of AM, we obtain a result of $4096 \times C$ one-hot encoded feature, in which C represents the number of semantic classes. The loss of semantic segmentation is calculated here.

2.6. Loss Functions. The loss functions play an indispensable role in the training of deep learning networks, and our PSegNet uses carefully designed different loss functions to supervise different tasks at the same time. In the semantic segmentation task, we use the standard cross-entropy loss function L_{sem}, which is defined as follows:

$$L_{\mathrm{sem}} = -\sum_{i=1}^{N}\sum_{j=1}^{C} x_j'(i) \log x_j(i), \tag{4}$$

in which $x_j(i)$ is the predicted probability that the current point $\mathbf{p}_i$ belongs to class j, and $x_j'(i)$ is the one-hot encoding of the true semantic class of the point. If the point truly belongs to the category j, the value of $x_j'(i)$ is 1, otherwise 0. In the instance segmentation task, the number of instances in an input point cloud is variable. Therefore, we use a comprehensive loss function that includes three weighted sublosses under an uncertain number of instances to supervise the training. The equation for the instance loss L_{ins} is given as follows:

$$L_{\mathrm{ins}} = \alpha \cdot L_s + \beta \cdot L_d + \gamma \cdot L_{\mathrm{reg}}. \tag{5}$$

The weights of sublosses $L_s, L_d, L_{\mathrm{reg}}$ in the total loss are represented by α, β, γ, respectively. L_s is devised to make it easier for the points of the same instance label to gather together. The function L_d is to make the points of different instance labels to repel each other in clustering to facilitate accurate instance segmentation. L_{reg} is the regularization loss which is used to make all cluster centers close to the origin of the feature space and help to form an effective and regular feature boundary for the embedding. The equations of L_s, L_d, L_{reg} are given as follows:

$$L_s = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{N_i}\sum_{j=1}^{N_i}\left[\max\left[0, \left\|\mathbf{c}_i - \mathbf{f}_j\right\|_2 - \delta_s\right]\right]^2, \quad (6)$$

$$L_d = \frac{1}{\binom{I}{2}}\sum_{i_A=1}^{I}\sum_{\substack{i_B=1 \\ i_B\neq i_A}}^{I}\left[\max\left[0, 2\delta_d - \left\|\mathbf{c}_{i_A} - \mathbf{c}_{i_B}\right\|_2\right]\right]^2, \quad (7)$$

$$L_{\text{reg}} = \frac{1}{I}\sum_{i=1}^{I}\left\|\mathbf{c}_i\right\|_2, \quad (8)$$

where I represents the number of instances in the current point cloud batch being processed and N_i represents the number of points contained in the i-th instance; $\mathbf{c}_i$ represents the center of the points belonging to the i-th instance in the current feature space, and $\mathbf{f}_j$ represents the feature vector of the point j in the current feature space. The parameter δ_s defines a boundary threshold that allows the aggregation of points of the same instance, and $2\delta_d$ represents the nearest feature distance threshold of two different instances.

In addition, in the output feature F_{DGF} of the DGFFM, in order to help integrating coarse-grained features and fine-grained features, it is also necessary to impose a supervision on this midlevel feature space. The purpose is to constrain the features belonging to the same instance to come closer in advance, while the features belonging to different instances to drift apart. Reference [69] proposed a point-level Double-hinge Loss (DHL) L_{DHL}, which considered the constraints of the semantic task and the instance task on the middle-level features of the network. We directly transplanted DHL to the feature map F_{DGF}; therefore, the total loss function of PSegNet can be represented as follows:

$$L = L_{\text{sem}} + L_{\text{ins}} + L_{\text{DHL}}. \quad (9)$$

2.7. Evaluation Measures. In order to verify the semantic segmentation performance of PSegNet on the plant point cloud, we calculate four quantitative measures—Prec, Rec, F1, and Intersection over Union (IoU) for each semantic class, respectively. For all the four semantic measures (represented in percentage), the higher means the better. The measure Prec means precision, and it is the proportion of the points correctly classified in this semantic class to all the points predicted by the network. The notation Rec means recall, which reflects the proportion of the points correctly classified in this semantic class to the total points of this class in the ground truth. IoU reflects the degree of overlapping between the predicted areas of each semantic category and the corresponding real areas, and F1 is a comprehensive indicator calculated as the harmonic average of Prec and Rec. The equations of the four quantitative measures are given in the first half of Table 2, in which TP represents the number of true positive points of the current semantic category, and FP represents the false positive points of the current category, while FN stands for the false negative points.

The measures—mCov, mWCov, mPrec, and mRec, were used to evaluate the results of instance segmentation, and the equations of the four measures are defined in the second half of Table 2, respectively. In Table 2, IG_m represents the ground truth point set of the m-th instance under the same semantic class; IP_n represents the predicted point set of the n-th instance under the same semantic class. max [·] represents the maximum value of all terms evaluated. The binary function IoU[·, ·], which accepts two inputs as the point set of ground truth and the predicted point set from the network, is calculated exactly as the semantic IoU equation. The parameter C is the number of semantic classes for calculation of the Mean Precision (mPrec) and the Mean Recall (mRec). Because the dataset has three plant species, the semantic classes include the stem class and the leaf class of each plant species, which fixes C at 6. The notation $|TP(\text{sem} = i)|$ represents the number of predicted instances whose IoU is above 0.5 in the semantic class i. The notation $|\text{IP}(\text{sem} = i)|$ represents the total predicted number of instances in semantic class i. $|\text{IG}(\text{sem} = i)|$ represents the number of instances of the ground truth in semantic class i.

Table 2: The quantitative measures used in this work.

	Measures	Equations
Semantic segmentation	Prec	$\frac{\text{TP}}{\text{TP} + \text{FP}}$
	Rec	$\frac{\text{TP}}{\text{TP} + \text{FN}}$
	F1	$2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$
	IoU	$\frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$
Instance segmentation	mCov	$\frac{1}{\text{I}}\sum_{m=1}^{I}\max_n[\text{IoU}[\text{IG}_m, \text{IP}_n]]$
	mWCov	$\sum_{m=1}^{I} w_m \max_n[\text{IoU}[\text{IG}_m, \text{IP}_n]]$
	mPrec	$\frac{1}{C}\sum_{i=1}^{C}\frac{\lvert\text{TP}(\text{sem} = i)\rvert}{\lvert\text{IP}(\text{sem} = i)\rvert}$
	mRec	$\frac{1}{C}\sum_{i=1}^{C}\frac{\lvert\text{TP}(\text{sem} = i)\rvert}{\lvert\text{IG}(\text{sem} = i)\rvert}$

3. Experiments and Results

3.1. Data Preparation and Training Details. The plant point cloud data used in this work originates from a laser-scanned point cloud dataset in [85, 86]. The dataset recorded three kinds of crops including tomato, tobacco, and sorghum growing in several different environments. Each crop was scanned multiple times during 30 days. We show the three crops of different growth periods in Figure 4. The scanning error of the dataset is controlled within ±25μm. The dataset contains a total of 546 individual point clouds including 312 tomato point clouds, 105 tobacco point clouds, and 129 sorghum point clouds. The largest point cloud contains more than 100000 points, and the smallest has about 10000 points.

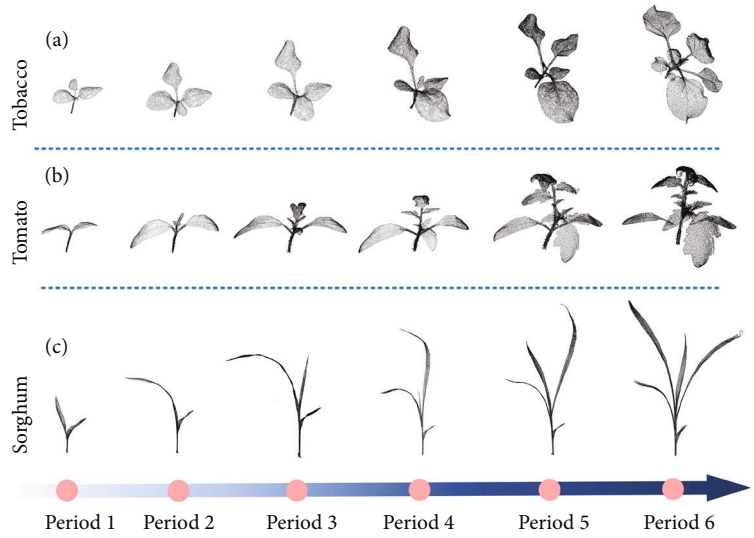

FIGURE 4: Demonstration of some point clouds from our dataset. (a) Point clouds in 6 consecutive growth periods of the same tobacco plant, respectively; (b) point clouds in 6 consecutive growth periods of the same tomato plant; (c) point clouds in consecutive 6 growth periods of the same sorghum plant.

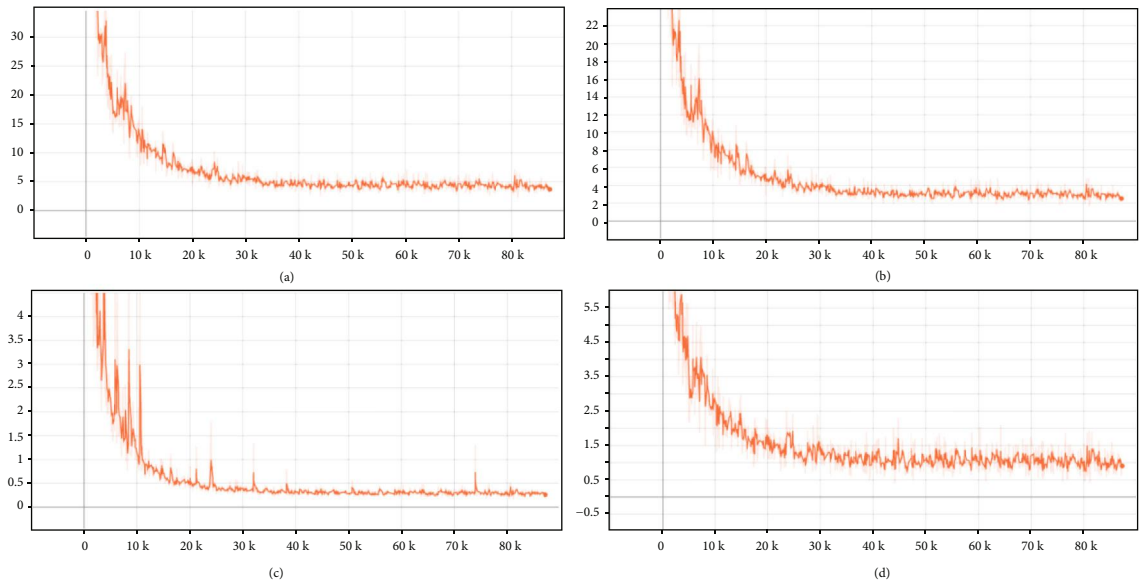

FIGURE 5: The changes of losses in the training of PSegNet. From (a–d) are the total loss L, the DHL L_{DHL} imposed on the midlevel feature layer after DGFMM, the semantic loss L_{sem}, and the instance loss L_{ins} . The x-axis of all plots means the number of trained samples, and the y-axis is the loss value. Given 3640 training samples and the training batch size at 8, we have 455 samples to be trained in each epoch. When the training stops at 190 epochs, the x-axis ends at $455 * 190 = 86450$.

We applied Semantic Segmentation Editor (SSE) [87] to annotate leaf and stem for semantic labels and the instance label for each single leaf. To be more specific, for semantic annotation, we classify the stem system and leaves of different species as different semantic categories. Therefore, in our dataset, there are six semantic categories with $C = 6$.

In order to prepare the data for network training and testing, the dataset should be divided and extended. Firstly, we divide the original dataset into the training set and test set under the ratio 2 : 1. The original dataset contains 546 point clouds, and for each point cloud, we used the VFPS method introduced in Section 2.1 to downsample it to a point cloud of $N = 4096$, and we repeated the downsampling for 10 times with randomly selected initial point in the last step of VFPS to augment the dataset. The randomness of data augmentation comes from the difference on the initial iteration point of FPS after voxelization in VFPS. Therefore, each point cloud in the dataset contains 4096 points after augmentation, and for the augmented 10 clouds generated from the same original point cloud, despite their similarity

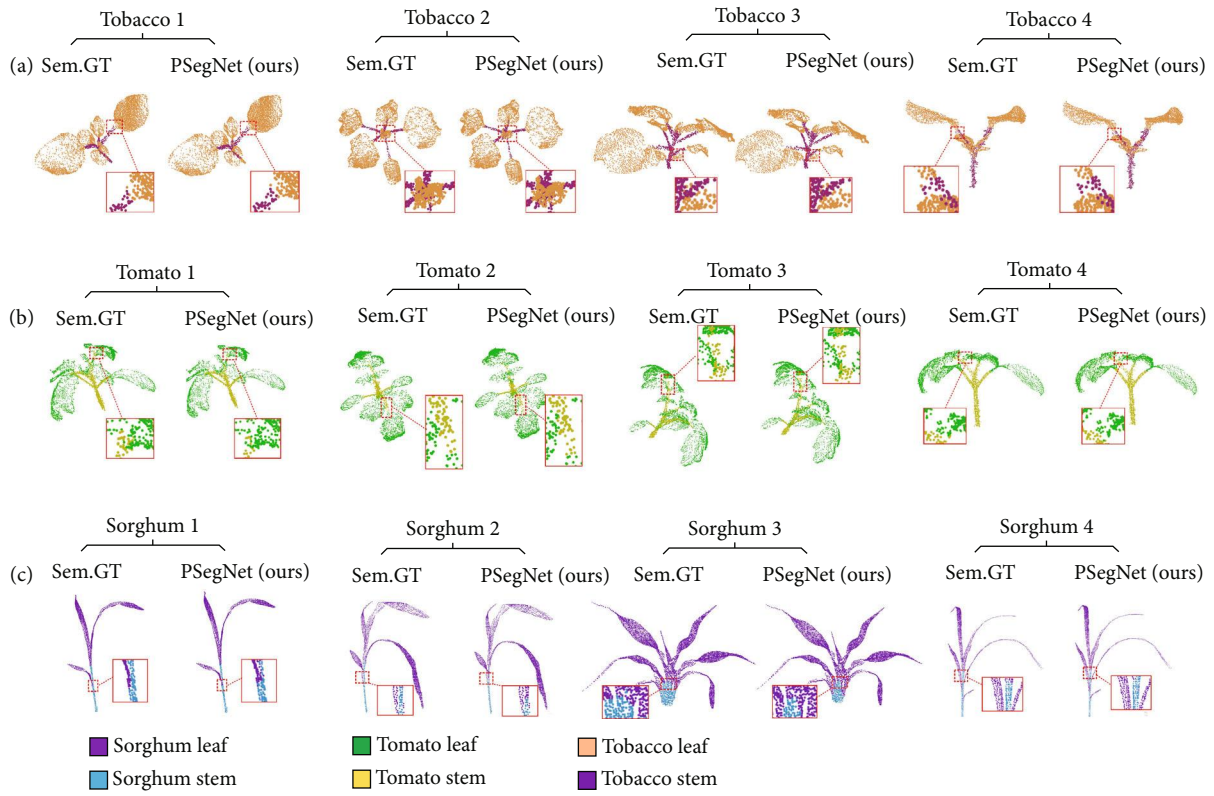

Figure 6: Qualitative demonstration of our PSegNet for semantic segmentation. (a) Semantic segmentation results of four different tobacco individuals, respectively. (b) Semantic segmentation results of four different tomato plants, respectively. (c) Semantic segmentation results of four different sorghum plants, respectively. Each segmented crop point cloud from PSegNet is compared with its corresponding ground truth (Sem.GT). The meanings of different rendered colors are shown at the bottom of the figure. Some of the areas are enlarged to give more details.

on appearance, the distributions of local points are quite different. Finally, we form the training set with 3640 point clouds and the test set with 1820 point clouds.

All the annotation work, training, testing, and the comparative experiments were conducted on a server under the Ubuntu 20.4 operating system. Our hardware platform contains an AMD RYZEN 2950x CPU that has 16 cores and 32 threads, a memory of 128 GB, and a GPU of NVIDIA RTX 2080Ti. The deep learning framework is TensorFlow 1.13.1. During training, batch size is fixed to 8, and the initial learning rate is set to 0.002; afterwards, the learning rate is reduced by 30% after 10 epochs per iteration. Adam solver is used to optimize our network, and Momentum is set to 0.9. For PSegNet and other methods compared, we end training at 190 epochs and record the model that has the minimum loss in testing as the selected model. In the encoder part of PSegNet, we set $k = 16$ in the first two DNFEBs and $k = 8$ for the last two DNFEBs for the KNN search range. The reasons of the KNN configuration are twofold. First, a large K brings a high calculation burden; therefore, K should not be a large number. Second, the number of features shrinks in the encoder part of PSegNet (from 4096 to 128) for encoding point cloud information efficiently. Thereby, the search range of the local KNN should also be declining with the shrinking features to keep the receptive field of the network stable. When building L_s and L_d, we set $\delta_s = 0.5$, and $\delta_d = 1.5$. We also fixed $\alpha = \beta = 1$ and $\gamma = 0.001$ throughout this study. We recoded the changes of losses of PSegNet during training; the values of the total loss and the three sublosses are displayed in Figure 5. All losses have shown an evident decline at first and a quick convergence.

Table 3: The quantitative measures of PSegNet on semantic segmentation.

	Tobacco		Tomato		Sorghum	
	Stem	Leaf	Stem	Leaf	Stem	Leaf
Prec (%)	92.71	96.76	96.36	97.98	89.54	98.04
Rec (%)	87.42	97.73	95.02	98.59	85.69	98.63
F1 (%)	89.99	97.24	95.68	98.29	87.57	98.33
IoU (%)	81.79	94.63	91.73	96.63	77.89	96.72

3.2. Semantic Segmentation Results. Figure 6 presents qualitative semantic results of PSegNet on three crop species. In order to reveal the real performance, we especially show test samples from different growth environments and stages. From Figure 6, we can see good segmentation results on all three crops. PSegNet seems to be good at determining the boundary between the stem and the leaves, because only rare

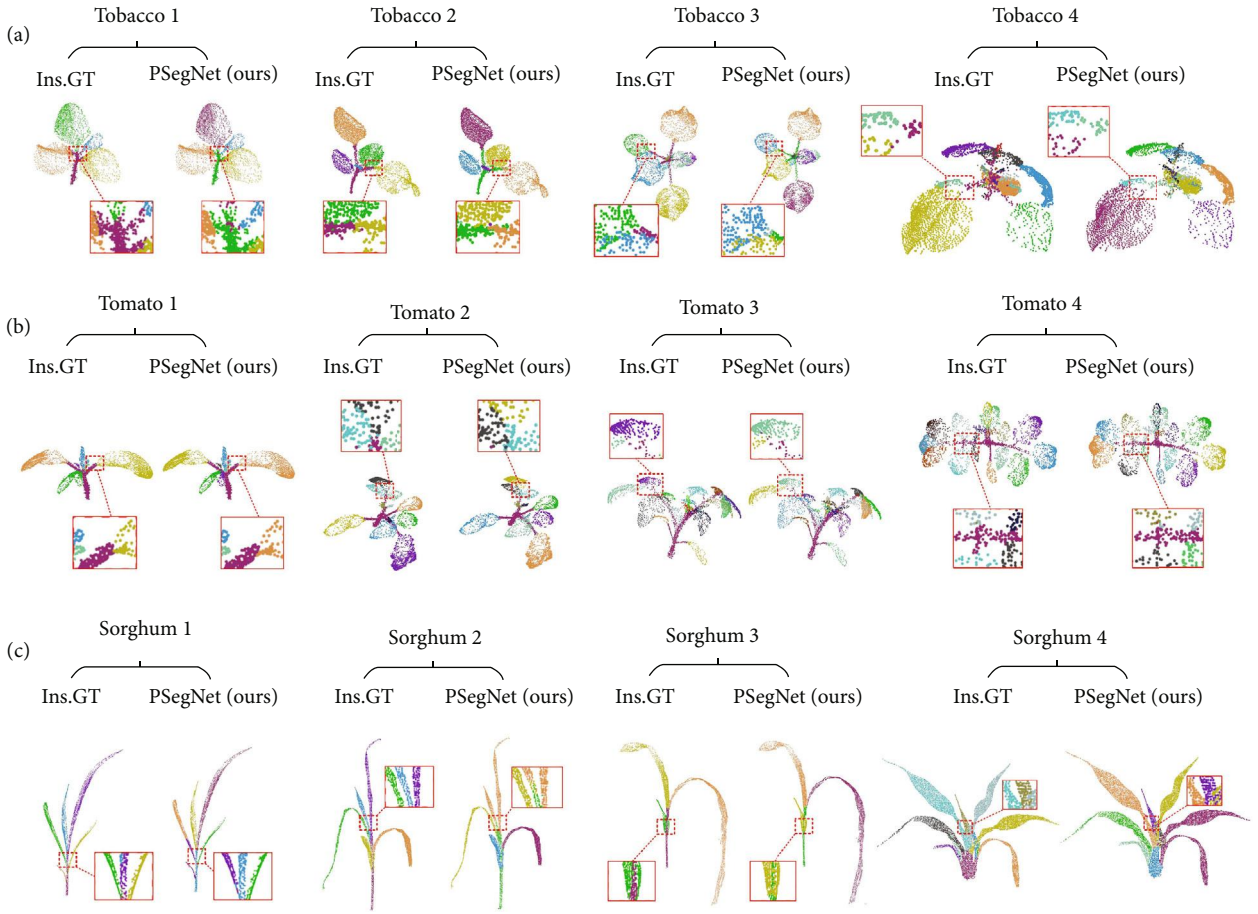

Figure 7: The qualitative demonstration of the instance segmentation by PSegNet. (a) Instance segmentation results of four different tobacco individuals, respectively. (b) Instance segmentation results of four different tomato individuals, respectively. (c) Instance segmentation results of four different sorghum individuals, respectively. Each segmented crop point cloud from PSegNet is compared with its corresponding ground truth (Ins.GT). Note that the different rendered colors in this figure are just for better visual separation of different instances, and it has no connection with the instance labels. Therefore, despite successful segmentation, the same leaf instance in the ground truth and the network result may be rendered with two different colors. Some of the areas are enlarged to give more details.

false segmentations can be observed around the boundary between the two semantic classes.

Table 3 presents the quantitative semantic segmentation results of PSegNet for the total test set, on which most measures have reached above 85.0%, showing satisfactory semantic segmentation performance. From the measures listed in Table 3, it is not hard to observe that the leaf segmentation results of PSegNet are better than the stem results, and this is because the point number of stems is much fewer than leaves in the training data. Across the three species, tomato has the best semantic segmentation result, and the possible reason is that the tomato point clouds account for the largest proportion in the total training dataset. This imbalance training can be improved by adding more data from the two other species.

3.3. Instance Segmentation Results. Figure 7 shows the qualitative evaluation of the instance results of three crops by PSegNet, and 12 representative point clouds from multiple growth stages are shown in Figure 7, respectively. Satisfactory segmentation of leaf instances can be observed on all three species in Figure 7. The three species differ heavily in leaf structure. Tobacco has big and broad leaves, and tomato plant has a compound leaf structure which contains at least one big leaflet and two small leaflets, while sorghum has long and slender leaves. PSegNet shows good leaf instance segmentation performance on all three types of leaves.

Table 4: The quantitative measures of PSegNet on instance segmentation.

	Tobacco leaf	Tomato leaf	Sorghum leaf
mPrec (%)	87.80	89.92	86.68
mRec (%)	77.09	78.62	82.13
mCov (%)	80.88	84.89	84.27
mWCov (%)	90.72	90.45	87.46

Table 4 lists the quantitative measures of leaf instance segmentation by PSegNet for the test set. Most of measures are above 80.0%, representing satisfactory instance segmentation performance.

Table 5: Quantitative comparison of semantic segmentation performances of six networks including PSegNet.

	Methods	Tobacco		Tomato		Sorghum		Mean
		Stem	Leaf	Stem	Leaf	Stem	Leaf	
Prec (%)	PointNet	77.15	94.02	93.99	96.71	77.87	95.37	89.19
	PointNet++	87.78	95.62	93.65	96.80	78.01	**98.33**	91.70
	ASIS	91.65	91.94	93.55	97.14	85.47	95.17	92.49
	PlantNet	89.45	**96.80**	95.90	96.30	89.07	97.43	94.16
	DGCNN	90.55	96.42	95.24	97.86	83.95	97.37	93.57
	PSegNet (ours)	**92.71**	96.76	**96.36**	**97.98**	**89.54**	98.04	**95.23**
Rec (%)	PointNet	79.20	93.31	91.85	97.61	61.45	97.85	86.88
	PointNet++	**90.83**	94.05	92.45	97.33	78.66	98.27	91.93
	ASIS	83.85	96.11	92.87	95.51	81.65	97.88	91.31
	PlantNet	86.12	92.97	**95.24**	98.23	**86.06**	98.07	92.78
	DGCNN	85.55	**97.76**	94.15	98.27	78.05	98.20	92.00
	PSegNet (ours)	87.42	97.73	95.02	**98.59**	85.69	**98.63**	**93.85**
F1 (%)	PointNet	78.16	93.66	92.91	97.16	68.69	96.59	87.86
	PointNet++	89.28	94.83	93.05	97.07	78.33	98.30	91.81
	ASIS	87.58	93.98	93.21	96.32	83.52	96.51	91.85
	PlantNet	87.75	94.85	95.56	97.26	87.54	97.75	93.45
	DGCNN	87.98	97.09	94.69	98.07	80.89	97.78	92.75
	PSegNet (ours)	**89.99**	**97.24**	**95.68**	**98.29**	**87.57**	**98.33**	**94.52**
IoU (%)	PointNet	64.15	88.08	86.76	94.48	52.31	93.41	79.87
	PointNet++	80.63	90.17	87.00	94.30	64.38	96.65	85.52
	ASIS	77.91	88.64	87.29	92.90	71.70	93.25	85.28
	PlantNet	78.17	90.20	91.51	94.66	77.84	95.59	88.00
	DGCNN	78.54	94.34	89.92	96.20	76.92	95.65	88.60
	PSegNet (ours)	**81.79**	**94.63**	**91.73**	**96.63**	**77.89**	**96.72**	**89.90**

The best values are in boldface.

3.4. Comparison with Other Methods. In this subsection, several mainstream point cloud segmentation networks are compared with our PSegNet on the same plant dataset. Among them, PointNet [64], PointNet++ [65], and DGCNN [74] are only capable of semantic segmentation. Like our network, ASIS [76] and PlantNet [77] can conduct the semantic segmentation and instance segmentation task simultaneously, and we use the same set of semantic and instance labels for training on the three dual-function networks. We used recommended parameter configurations for the comparative networks from their original papers, respectively.

Table 5 shows the quantitative comparison across six networks including PSegNet on the semantic segmentation task. PSegNet achieved the best in most cases and was superior to the others on all four averaged quantitative measures. Table 6 shows the quantitative performance comparison of PSegNet with two dual-function segmentation networks (ASIS and PlantNet) on instance segmentation. Except for the mPrec, mRec, and mCov of sorghum leaves, our PSegNet has achieved the best performance at all cases including all the four averaged measures.

We also compared PSegNet with state of the art qualitatively on both semantic segmentation and instance segmentation tasks. The qualitative semantic segmentation comparison on the three species is shown in Figure 8, and the instance segmentation comparison is shown in Figure 9. The samples in the two figures exhibit that PSegNet is superior to the networks that are specially designed for semantic segmentation, and PSegNet is also superior to the dual-function segmentation networks—ASIS and PlantNet for instance segmentation.

3.5. Ablation Study. In this section, we designed several independent ablation experiments to verify the effectiveness of the proposed modules in PSegNet, including VFPS, DNFEB, and DGFFM, as well as the SA and CA in the AM. The ablation experiments on semantic segmentation are shown in Table 7, and the ablation experiments on instance segmentation are shown in Table 8. In the two tables, the "Ver" column gives the version names of the ablated networks, respectively. Each version is formed by removing an existing module or some parts of a module from the original PSegNet. We compared seven versions of PSegNet named "A1" to "A7" with the complete PSegNet (named with "C"). In order to ensure the fairness of the experiments, ablating VFPS (A6) means using the basic FPS for downsampling and augmentation of the point cloud data. After ablating a module with convolutions, we will add MLPs with the same depth at the

Table 6: Quantitative comparison of instance segmentation performances of six networks including PSegNet.

	Methods	Tobacco leaf	Tomato leaf	Sorghum leaf	Mean
mPrec (%)	ASIS	78.54	80.21	79.04	79.26
	PlantNet	87.74	85.50	79.39	84.21
	PSegNet (ours)	**87.80**	**89.92**	**86.68**	**88.13**
mRec (%)	ASIS	56.27	64.84	72.08	64.40
	PlantNet	69.36	76.63	81.83	75.94
	PSegNet (ours)	**77.09**	**78.62**	**82.13**	**79.28**
mCov (%)	ASIS	62.88	76.61	74.26	71.25
	PlantNet	71.98	83.34	82.63	79.32
	PSegNet (ours)	**80.88**	**84.89**	**84.27**	**83.35**
mWCov (%)	ASIS	73.95	82.73	77.31	78.00
	PlantNet	84.83	89.48	85.68	86.66
	PSegNet (ours)	**90.72**	**90.45**	**87.46**	**89.54**

The best values are in boldface.

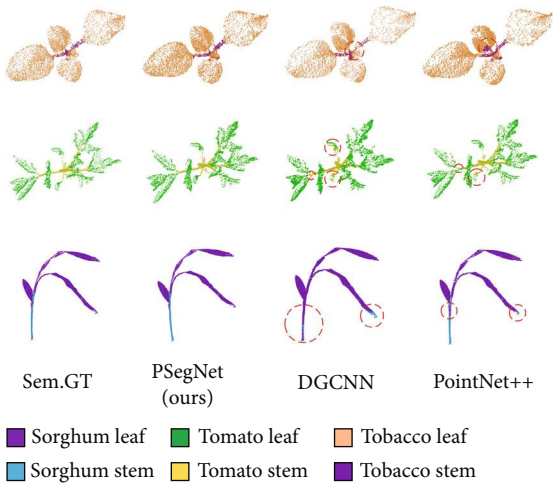

Figure 8: The qualitative semantic segmentation comparison on the three species. DGCNN and PointNet++ are compared with our PSegNet. The parts with segmentation errors are highlighted by red dotted circles, respectively. DGCNN and PointNet++ both have multiple prediction errors around the boundary between two different point classes.

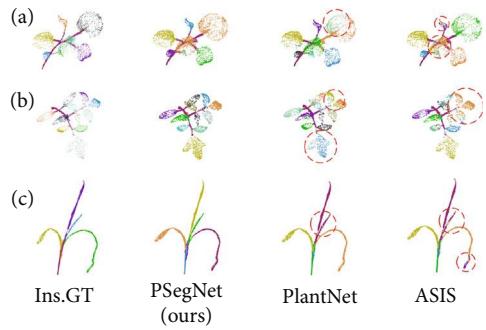

Figure 9: The qualitative instance segmentation comparison on the three species. (a) The tobacco plant and (b) the tomato plant; (c) the sorghum plant. PlantNet and ASIS are compared with our PSegNet. Note that the different rendered colors in this figure are just for better visual effect, and the colors are not associated with the instance labels. The parts with segmentation errors are highlighted by red dotted circles, respectively. PlantNet and ASIS both have multiple prediction errors around the boundaries of leaf instances.

ablated position to ensure that the network depth (or the number of parameters) remains unchanged. For example, when ablating DNFEB, we replace it with a 6-layer MLP to form the A5 network. When ablating SA, we replace it with a 1-layer MLP to form the A2 network. When ablating DGFFM (A4), only the decoder branch that extracts fine-grained features is left to keep the depth of the network unchanged. In the A7 network, we only keep the stage 1 for all DNFEBs in PSegNet to validate the effectiveness of the 3-stage structure of DNFEB in feature extraction. In Table 7, the complete version of PSegNet has the best semantic segmentation performance in average, which proves the ablation of any proposed submodule will lead to the decline on the average segmentation performance and also indirectly verifies the effectiveness of all proposed submodules and the sampling strategy in the paper. In the ablation analysis experiments shown in Table 8, the complete PSegNet network has the best average instance segmentation performance. Ablating any proposed submodule will also lead to the decline of the average network segmentation performance, which again verifies the effectiveness of all proposed submodules.

4. Discussion

4.1. Generalization Ability of PSegNet. In this subsection, we prove that the proposed PSegNet is versatile enough to be applied to other types of point cloud data, not only restricted to 3D datasets for plant phenotyping. We trained and tested our PSegNet on the *Stanford Large-Scale 3D Indoor Spaces* (S3DIS) [88] to validate its applicability on point clouds of

Table 7: The ablation analysis of PSegNet on semantic segmentation. The check mark stands for the use of a module. The best quantitative values are shown in bold. The sign "○" means the partial ablation of DNFEB.

	Ver	VFPS	DNFEB	DGFFM	MA		Tobacco		Tomato		Sorghum		Mean
					SA	CA	Stem	Leaf	Stem	Leaf	Stem	Leaf	
Prec (%)	A1	√	√	√	√		92.47	97.09	95.82	96.78	86.38	97.56	94.35
	A2	√	√	√		√	91.01	95.68	96.12	97.80	87.63	96.89	94.19
	A3	√	√	√			91.39	96.65	95.90	96.89	**90.31**	96.77	94.65
	A4	√	√		√	√	89.48	**97.29**	95.95	97.54	86.63	95.79	93.78
	A5	√		√	√	√	59.58	76.59	84.63	90.35	62.95	84.87	76.50
	A6		√	√	√	√	89.46	97.12	96.17	95.82	85.82	96.49	93.48
	A7	√	○	√	√	√	86.43	97.07	94.65	97.47	88.44	97.18	93.54
	C	√	√	√	√	√	**92.71**	96.76	**96.36**	**97.98**	89.54	**98.04**	**95.23**
Rec (%)	A1	√	√	√	√		**88.00**	94.46	**95.41**	98.24	**87.59**	97.58	93.55
	A2	√	√	√		√	86.68	96.24	94.79	98.04	84.26	98.28	93.05
	A3	√	√	√			87.54	92.69	95.20	98.11	87.36	**98.90**	93.30
	A4	√	√		√	√	85.88	94.23	94.49	**98.60**	85.22	97.90	92.72
	A5	√		√	√	√	59.18	77.13	81.02	90.59	52.10	88.49	74.75
	A6		√	√	√	√	87.49	90.02	94.95	98.59	84.56	97.69	92.22
	A7	√	○	√	√	√	82.86	95.26	94.21	98.47	85.48	97.98	92.38
	C	√	√	√	√	√	87.42	**97.73**	95.02	98.59	85.69	98.63	**93.85**
F1 (%)	A1	√	√	√	√		**90.18**	95.75	95.61	97.51	86.98	97.57	93.93
	A2	√	√	√		√	88.79	95.96	95.45	97.92	85.91	97.58	93.60
	A3	√	√	√			89.42	94.63	95.55	97.50	**88.81**	97.82	93.96
	A4	√	√		√	√	87.64	95.74	95.22	98.07	85.92	96.84	93.24
	A5	√		√	√	√	59.38	76.86	82.79	90.47	57.01	86.64	75.53
	A6		√	√	√	√	88.46	93.44	95.55	97.18	85.19	97.09	92.82
	A7	√	○	√	√	√	84.61	96.16	94.43	97.97	86.94	97.58	92.95
	C	√	√	√	√	√	89.99	**97.24**	**95.68**	**98.29**	87.57	**98.33**	**94.52**
IoU (%)	A1	√	√	√	√		**82.11**	91.85	91.60	95.13	76.96	95.25	88.82
	A2	√	√	√		√	79.84	92.23	91.30	95.92	75.30	95.27	88.31
	A3	√	√	√			80.87	89.81	91.47	95.12	**79.87**	95.74	88.81
	A4	√	√		√	√	78.01	91.82	90.87	96.21	75.32	93.87	87.68
	A5	√		√	√	√	42.22	62.42	70.63	82.60	39.87	76.43	62.36
	A6		√	√	√	√	79.31	87.68	91.48	94.52	74.20	94.34	86.92
	A7	√	○	√	√	√	73.32	92.60	89.44	96.02	76.89	95.27	87.26
	C	√	√	√	√	√	81.79	**94.63**	**91.73**	**96.63**	77.89	**96.72**	**89.90**

indoor scenes, which are very different from crops in 3D. The S3DIS dataset has 6 large indoor areas scanned by Lidars, including 271 rooms functioning as conference rooms, offices, and hallways. All points in the dataset are divided into 13 semantic classes such as floor, table, window, and so on. In addition, all points in each semantic class are labeled with instance indices. In training and testing, we cut each room into many $1 \times 1 \times$ he blocks measured in meter that do not overlap each other, and he means the height of each room. Each block was downsampled to 4096 points as a single point cloud input. The point clouds in Area 5 of S3DIS were used for testing, and the rest of the S3DIS areas were used for training. During training, we set all hyperparameters of PSegNet to be the same as the way the plant dataset was trained. Figure 10 shows the qualitative semantic segmentation results of several S3DIS rooms, respectively. The majority of points were correctly classified by PSegNet comparing to the GT, and our network seems to be especially good at recognizing furniture such as tables and chairs with varied shapes and orientations. Instance segmentation on S3DIS is regarded as a challenging task; however, our PSegNet shows satisfactory instance segmentation on the four different rooms in Figure 11. PSegNet seems to have better instance segmentation on small objects than large objects; e.g., in the third room, PSegNet almost correctly labeled all single chairs.

4.2. Discussion of the Effectiveness. In this subsection, we would like to explain more about how this work handles the three challenges faced by current deep learning models for point cloud segmentation.

Table 8: The ablation analysis of PSegNet on instance segmentation. The check mark stands for the use of a module. The best quantitative values are shown in bold. The sign "○" means the partial ablation of DNFEB.

	Ver	VFPS	DNFEB	DGFFM	MA		Tobacco leaf	Tomato leaf	Sorghum leaf	Mean
					SA	CA				
mPrec (%)	A1	√	√	√	√		86.98	88.60	77.53	84.37
	A2	√	√	√		√	86.44	90.50	82.28	86.41
	A3	√	√	√			90.00	88.07	79.98	86.02
	A4	√	√		√	√	87.93	89.02	77.78	84.91
	A5	√		√	√	√	64.19	76.23	38.73	59.72
	A6		√	√	√	√	**89.54**	89.08	79.72	86.11
	A7	√	○	√	√	√	87.83	88.19	83.64	86.55
	C	√	√	√	√	√	87.80	**89.92**	**86.68**	**88.13**
mRec (%)	A1	√	√	√	√		73.65	79.23	81.29	78.06
	A2	√	√	√		√	74.57	77.44	79.31	77.11
	A3	√	√	√			73.09	**79.41**	80.30	77.60
	A4	√	√		√	√	73.12	77.72	77.13	75.99
	A5	√		√	√	√	43.16	41.18	36.73	40.36
	A6		√	√	√	√	70.50	74.88	78.42	74.60
	A7	√	○	√	√	√	75.46	77.83	80.00	77.76
	C	√	√	√	√	√	**77.09**	78.62	**82.13**	**79.28**
mCov (%)	A1	√	√	√	√		76.73	85.06	82.49	81.43
	A2	√	√	√		√	78.87	84.34	81.80	81.67
	A3	√	√	√			75.38	**85.34**	83.11	81.28
	A4	√	√		√	√	77.27	84.20	81.75	81.07
	A5	√		√	√	√	44.50	58.10	42.09	48.23
	A6		√	√	√	√	74.83	82.51	81.61	79.65
	A7	√	○	√	√	√	77.51	84.57	81.21	81.10
	C	√	√	√	√	√	**80.88**	84.89	**84.27**	**83.35**
mWCov (%)	A1	√	√	√	√		87.72	90.24	86.31	88.09
	A2	√	√	√		√	89.77	89.78	86.00	88.52
	A3	√	√	√			86.85	90.06	86.48	87.80
	A4	√	√		√	√	87.92	89.60	85.32	87.61
	A5	√		√	√	√	64.04	71.86	49.17	61.69
	A6		√	√	√	√	84.29	88.64	85.67	86.20
	A7	√	○	√	√	√	87.67	89.52	84.47	87.22
	C	√	√	√	√	√	**90.72**	**90.45**	**87.46**	**89.54**

4.2.1. Why the proposed VFPS strategy prevails? To better understand this, we constructed a simple 2D lattice with only 18 points to simulate a flat leaf in space and compared the difference between FPS and VFPS on the lattice. The 2D lattice, shown in Figure 12, was intentionally set to be sparse at the upper part while dense at the lower part. The aim of sampling for both FPS and VFPS is the same, to reduce the number of points to only 7. We fixed the number of voxels in VFPS as 8, which was slightly larger than the aim of downsampling ($N = 7$) according to the instruction of VFPS. Figure 12(a) shows the whole process of FPS starting from the center point of the lattice, and in the sampled 7 points, only one of them is from the interior part, and the other 6 are edge points. Figure 12(c) shows FPS starting from the leftmost point, and all 7 sampled points are located on the edge of lattice. Figures 12(a) and 12(c) reflect a common phenomenon of FPS that when $n \gg N$, the sampling may concentrate on edges and easily create cavities on point clouds. And when $n \gg N$, FPS also deviates from the common intuition that the low-density area gets points easier than the high-density area because the upper part of lattice of Figure 12(c) is not getting more points. Figures 12(b) and 12(d) show two different processes of VFPS on the voxelized lattice initialized with the center point and the leftmost point, respectively. The two downsampled VFPS results are smoother than the counterparts of FPS and have smaller cavities inside. In a real point cloud of crop, the leaves are usually flat and thin, presenting a similar

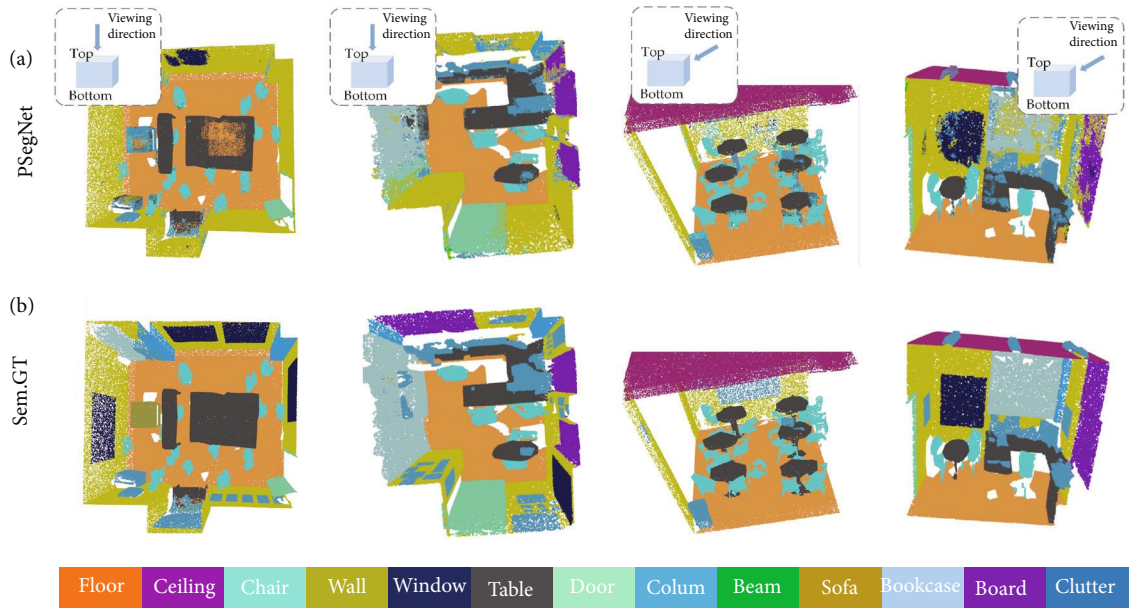

FIGURE 10: Demonstration of the semantic segmentation results of PSegNet on four different rooms in Area 5 of S3DIS. (a) The semantc prediction results from PSegNet and (b) the semantic ground truth. Different semantic classes are rendered with different colors at the bottom, respectively. The first two rooms are visualized from top, and the third and the fourth rooms are visualized with side views. Each room is composed of several $1 \times 1 \times$ he blocks.

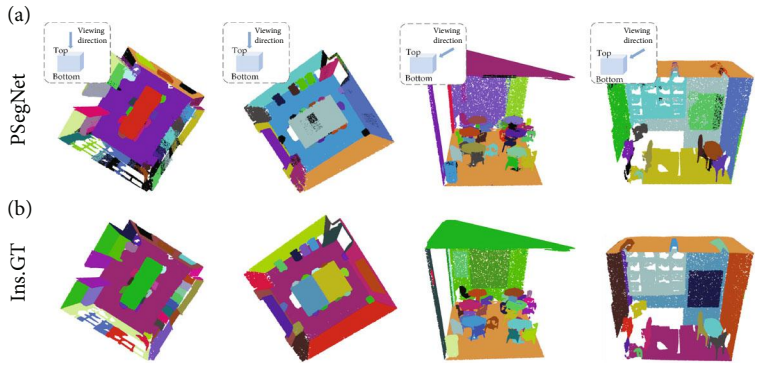

FIGURE 11: Demonstration of the instance segmentation results of PSegNet on four different rooms in Area 5 of S3DIS. (a) The instance prediction results from PSegNet and (b) the instance GT. Different instance classes are rendered with different colors, respectively. Note that the different rendered colors in this figure are just for better visual effect, and the colors are not associated with the instance labels. The first two rooms are visualized from top, and the third and the fourth rooms are visualized with side views. Each room is composed of several $1 \times 1 \times$ he blocks.

structure as Figure 12 lattice in the 3D space. Moreover, we are frequently challenged with the sampling requirement of $n \gg N$ in 3D plant phenotyping, on which VPFS can generate smooth results with smaller cavities.

4.2.2. How PSegNet strikes the balance between semantic segmentation and instance segmentation? The network design for multifunctional point cloud segmentation is difficult. The reasons are twofold. First, each single segmentation task needs a specially designed network branch controlled by a unique loss function. Take PSegNet as the example, the semantic segmentation pathway and the instance segmentation pathway are restricted by L_{sem} and L_{ins}, respectively. To better reconcile the training on the main network, we also added the point-level loss L_{DHL} to the feature map F_{DGF}. Therefore, the total network of PSegNet is restricted by a combination of three losses. The second difficulty in design is that when adjusting the weight of a branch's loss in the total loss, the other branch will also be effected. For example, when increasing the proportion of the semantic loss in the PSegNet, the instance performance will likely drop. Thus, the balance between the semantic segmentation and the instance segmentation can be reached by controlling the assigned weights to their losses, respectively. Fortunately, after several tests, we found that the proportion of 1 : 1 for L_{sem} and L_{ins} was a choice good enough to outcompete state of the art.

4.2.3. Why PSegNet has good generalization ability on plant species? For a point cloud learning model, the bad

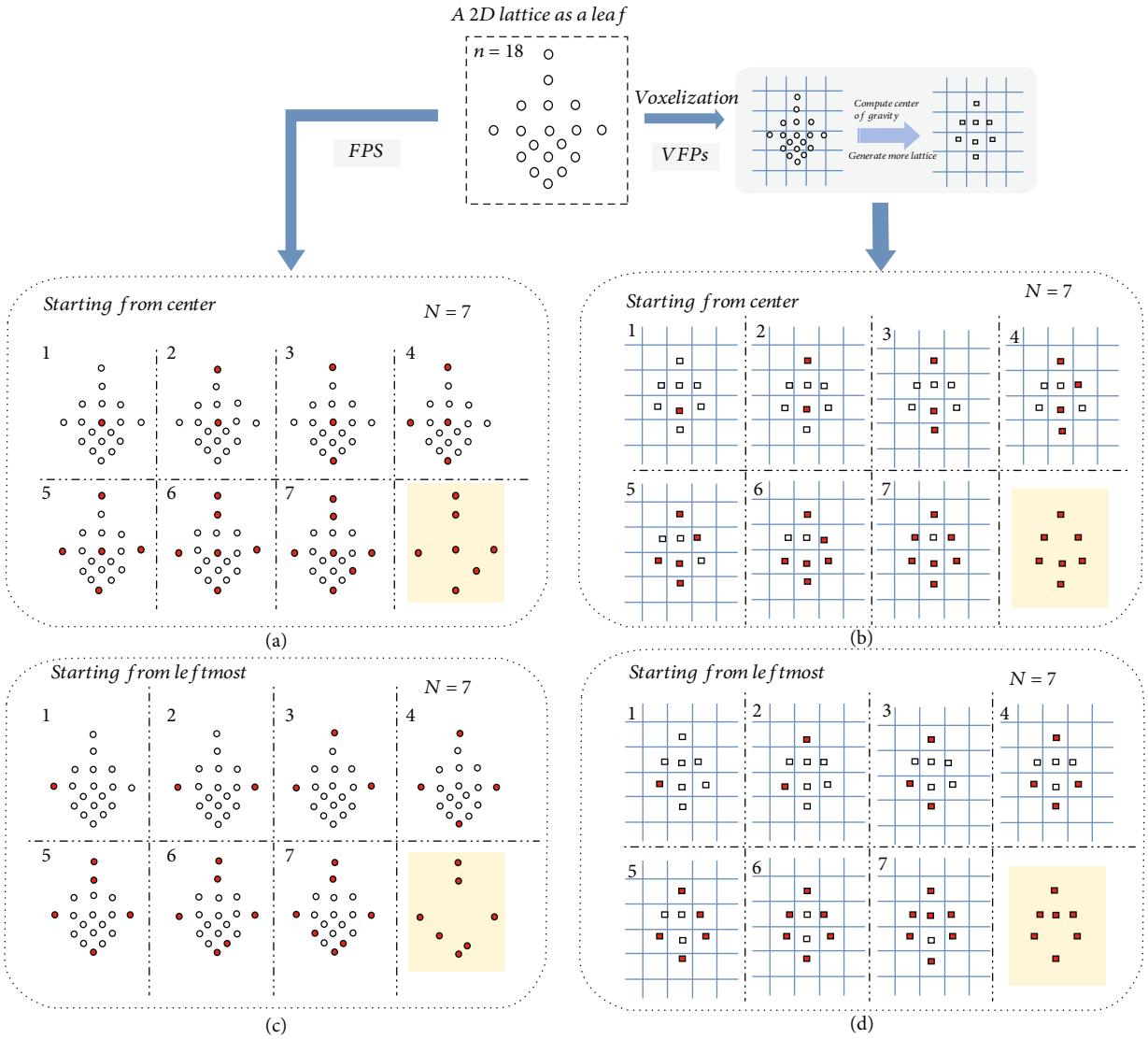

Figure 12: Comparison of VFPS and FPS on a 2D lattice like a leaf. The original lattice contains 18 points, sparse on the upper part and dense at the lower part. (a) The FPS ($N = 7$) process starting from the center; (b) the FPS ($N = 7$) process starting from the leftmost point; (c) VFPS ($N = 7$) process starting from the center point; (d) VFPS ($N = 7$) process starting from the leftmost point.

generalization on species usually results in two undesirable phenomena. The first is that one may observe parts of plant species "A" on the point cloud of a plant species "B"; e.g., the model may falsely classify some points on the stem of a tomato plant as "the stem of a tobacco" in semantic segmentation. The second phenomenon is that one may see a big gap on segmentation performance between the monocotyledons (sorghum) and the dicotyledons (tomato and tobacco) due to the huge differences in 3D structure. In the qualitative and quantitative results of PSegNet, the two undesirable phenomena were rarely seen, and PSegNet outperformed popular networks on both semantic and instance segmentation tasks. In addition, we also have found that PSegNet has strong generalization ability on the object types in point clouds. The test of PSegNet on indoor S3DIS dataset (given in Section 4.1) proved that the network has potential to be generalized to other fields such as indoor SLAM (Simultaneous Localization and Mapping) and Self-Driving Vehicles.

The good generalization ability of PSegNet is most likely to come from the design of the network architecture in Figure 2. The DNFEB separately extracts high-level and low-level features from two local spaces and then aggregates them to realize better learning. DGFFM first uses two parallel decoders to construct feature layers with two different learning granularities, respectively. Then, DGFFM fuses the two feature granularities to create comprehensive feature learning. The AM part of PSegNet uses two types of attentions (spatial and channel) to lift the network training efficiency on both segmentation tasks.

4.3. Limitations. Although PSegNet can perform two kinds of segmentations well on three species with several growth periods, this network still does not work well on seedlings.

For many plant species, the seedling period takes a very small and special 3D shape, which is different from all the other growth stages. Hence, the distinctiveness of the seedling period may cause problem in training and testing for deep learning networks.

The segmentation performance of PSegNet will decrease with the increasing complexity of the plant structure (e.g., trees), and unfortunately, this happens to all such networks. The reasons are twofold. First, the current dataset does not include plant samples with a lot of leaves; therefore, the network cannot work directly on samples of trees. Second, due to the restriction of hardware, the network only accepts 4096 points as one sample input. For a plant point cloud with dense foliage, the number of points on each organ will be very few, which causes sparse and bad feature learning and definitely outputs terrible segmentation results.

5. Conclusion

In this paper, we first proposed a Voxelized Farthest Point Sampling (VFPS) strategy. This new downsampling strategy for point clouds inherits merits from both the traditional FPS downsampling strategy and the VBS strategy. It can not only fix the number of points after downsampling but also able to augment the dataset with randomness, which renders it especially suitable for the training and testing of deep learning networks. Secondly, this paper designs a novel dual-function segmentation network—PSegNet; it is suitable for laser-scanned crop point clouds of multiple species. The end-to-end PSegNet is mainly composed of three parts—the front part has a typical encoder-like structure that is frequently observed in deep neural networks; the middle part is the Double-Granularity Feature Fusion Module (DGFFM), which decodes and integrates two features with different granularities. The third part of PSegNet has a double-flow structure with Attention Modules (AMs), in which the upper branch and the lower branch correspond to the instance segmentation task and semantic segmentation task, respectively. In qualitative and quantitative comparisons, our PSegNet outcompeted several popular networks including PointNet, PointNet++, ASIS, and PlantNet on both organ semantic and leaf instance segmentation tasks.

In the future, we will focus on two aspects. First, we will collect more crop point cloud data with high-precision and introduce more species (especially the monocotyledonous plants with slender organs) into the dataset as well. Secondly, we will devise new deep learning architectures that are more suitable for the understanding and processing of 3D plant structures and propose compressed networks with high segmentation accuracies to serve the real-time need in some scenarios of the agriculture industry.

Data Availability

The data and the code are available upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Authors' Contributions

D. L., J. L., and A. P. wrote the paper and crafted all figures and tables. D. L. and J. L. designed the whole architecture of PSegNet. J. L. carried out the experiments. S. X. prepared the dataset. J. L. and A. P. organized the references. All authors read and approved the final manuscript.

Acknowledgments

This work was supported in part by the Shanghai Rising-Star Program (No. 21QA1400100), Shanghai Natural Science Foundation (No. 20ZR1400800), Shanghai Sailing Program (No. 21YF1401400), and in part by Fundamental Research Funds for the Central Universities of China (No. 2232020D-49).

References

[1] Z. Li, R. Guo, M. Li, Y. Chen, and G. Li, "A review of computer vision technologies for plant phenotyping," *Computers and Electronics in Agriculture*, vol. 176, article 105672, 2020.

[2] P. Trivedi, *Advances in Plant Physiology*, IK International Pvt Ltd, 2006.

[3] T. Ogura and W. Busch, "Genotypes, networks, phenotypes: moving toward plant systems genetics," *Annual Review of Cell and Developmental Biology*, vol. 32, no. 1, pp. 103–126, 2016.

[4] H. J. Liu and J. Yan, "Crop genome-wide association study: a harvest of biological relevance," *The Plant Journal*, vol. 97, no. 1, pp. 8–18, 2019.

[5] C. Costa, U. Schurr, F. Loreto, P. Menesatti, and S. Carpentier, "Plant phenotyping research trends, a science mapping approach," *Frontiers in Plant Science*, vol. 9, p. 1933, 2019.

[6] L. Feng, M. A. Raza, Z. Li et al., "The influence of light intensity and leaf movement on photosynthesis characteristics and carbon balance of soybean," *Frontiers in Plant Science*, vol. 9, 2019.

[7] T. W. Gara, A. K. Skidmore, R. Darvishzadeh, and T. Wang, "Leaf to canopy upscaling approach affects the estimation of canopy traits," *GIScience & remote sensing*, vol. 56, no. 4, pp. 554–575, 2019.

[8] L. Fu, E. Tola, A. Al-Mallahi, R. Li, and Y. Cui, "A novel image processing algorithm to separate linearly clustered kiwifruits," *Biosystems Engineering*, vol. 183, pp. 184–195, 2019.

[9] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, "Mango leaf image segmentation on HSV and YCbCr color spaces using Otsu thresholding," in *2017 3rd International Conference on Science and Technology-Computer (ICST)*, pp. 99–103, Yogyakarta, Indonesia, 2017.

[10] Z. Wang, K. Wang, F. Yang, S. Pan, and Y. Han, "Image segmentation of overlapping leaves based on Chan–Vese model and Sobel operator," *Information processing in agriculture*, vol. 5, no. 1, pp. 1–10, 2018.

[11] R. Thendral, A. Suhasini, and N. Senthil, "A comparative analysis of edge and color based segmentation for orange

fruit recognition," in *2014 International Conference on Communication and Signal Processing*, pp. 463–466, Melmaruvathur, India, 2014.

[12] H. Scharr, M. Minervini, A. P. French et al., "Leaf segmentation in plant phenotyping: a collation study," *Machine Vision and Applications*, vol. 27, no. 4, pp. 585–606, 2016.

[13] P. Deepa and S. Geethalakshmi, "Improved watershed segmentation for apple fruit grading," in *2011 International Conference on Process Automation Control and Computing*, pp. 1–5, Coimbatore, India, 2011.

[14] A. Abinaya and S. M. M. Roomi, "Jasmine flower segmentation: a superpixel based approach," in *2016 international Conference on Communication and Electronics Systems (ICCES)*, pp. 1–4, Coimbatore, India, 2016.

[15] X. Niu, M. Wang, X. Chen, S. Guo, H. Zhang, and D. He, "Image segmentation algorithm for disease detection of wheat leaves," in *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pp. 270–273, Kumamoto, Japan, 2014.

[16] J. C. Neto, G. E. Meyer, and D. D. Jones, "Individual leaf extractions from young canopy images using Gustafson–Kessel clustering and a genetic algorithm," *Computers and Electronics in Agriculture*, vol. 51, no. 1-2, pp. 66–85, 2006.

[17] J.-M. Pape and C. Klukas, *3-D Histogram-Based Segmentation and Leaf Detection for Rosette Plants*, European Conference on Computer Vision. Springer, Cham, 2014.

[18] X. Yin, X. Liu, J. Chen, and D. M. Kramer, "Multi-leaf alignment from fluorescence plant images," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 437–444, Steamboat Springs, CO, USA, 2014.

[19] C. Kalyoncu and Ö. Toygar, "Geometric leaf classification," *Computer Vision and Image Understanding*, vol. 133, pp. 102–109, 2015.

[20] J.-M. Pape and C. Klukas, "Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images," in *Procedings of the Proceedings of the Computer Vision Problems in Plant Phenotyping Workshop 2015*, pp. 1–12, Swansea, UK, 2015.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, 2015.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Venice, Italy, 2017.

[24] N. Sapoukhina, S. Samiei, P. Rasti, and D. Rousseau, "Data augmentation from RGB to chlorophyll fluorescence imaging application to leaf segmentation of Arabidopsis thaliana from top view images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2563–2570, Long Beach, CA, USA, 2019.

[25] D. Ward, P. Moghadam, and N. Hudson, "Deep leaf segmentation using synthetic data," 2018, https://arxiv.org/abs/1807.10931.

[26] P. Sadeghi-Tehran, N. Virlet, E. M. Ampe, P. Reyns, and M. J. Hawkesford, "DeepCount: in-field automatic quantification of wheat spikes using simple linear iterative clustering and deep convolutional neural networks," *Frontiers in Plant Science*, vol. 10, 2019.

[27] J. P. Kumar and S. Domnic, "Image based leaf segmentation and counting in rosette plants," *Information Processing in Agriculture*, vol. 6, no. 2, pp. 233–246, 2019.

[28] S. Aich and I. Stavness, "Leaf counting with deep convolutional and deconvolutional networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2080–2089, Venice, Italy, 2017.

[29] B. Romera-Paredes and P. H. S. Torr, *Recurrent Instance Segmentation*, European conference on computer vision. Springer, Cham, 2016.

[30] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Transactions on Graphics*, vol. 29, no. 6, pp. 1–8, 2010.

[31] Z. Koma, M. Rutzinger, and M. Bremer, "Automated segmentation of leaves from deciduous trees in terrestrial laser scanning point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1456–1460, 2018.

[32] S. Jin, Y. Su, F. Wu et al., "Stem–leaf segmentation and phenotypic trait extraction of individual maize using terrestrial LiDAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1336–1346, 2019.

[33] W. Su, M. Zhang, J. Liu et al., "Automated extraction of corn leaf points from unorganized terrestrial LiDAR point clouds," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 3, pp. 166–170, 2018.

[34] S. Sun, C. Li, and A. H. Paterson, "In-field high-throughput phenotyping of cotton plant height using LiDAR," *Remote Sensing*, vol. 9, no. 4, p. 377, 2017.

[35] J. A. Jimenez-Berni, D. M. Deery, P. Rozas-Larraondo et al., "High throughput determination of plant height, ground cover, and above-ground biomass in wheat with LiDAR," *Frontiers in Plant Science*, vol. 9, p. 237, 2018.

[36] Q. Guo, F. Wu, S. Pang et al., "Crop 3D—a LiDAR based platform for 3D high-throughput crop phenotyping," *Science China Life Sciences*, vol. 61, no. 3, pp. 328–339, 2018.

[37] H. Yuan, R. S. Bennett, N. Wang, and K. D. Chamberlin, "Development of a peanut canopy measurement system using a ground-based lidar sensor," *Frontiers in Plant Science*, vol. 10, p. 203, 2019.

[38] A. Vit and G. Shani, "Comparing rgb-d sensors for close range outdoor agricultural phenotyping," *Sensors*, vol. 18, no. 12, p. 4413, 2018.

[39] X. Wang, D. Singh, S. Marla, G. Morris, and J. Poland, "Field-based high-throughput phenotyping of plant height in sorghum using different sensing technologies," *Plant Methods*, vol. 14, no. 1, pp. 1–16, 2018.

[40] C. Wang, X. Zou, Y. Tang, L. Luo, and W. Feng, "Localisation of litchi in an unstructured environment using binocular stereo vision," *Biosystems Engineering*, vol. 145, pp. 39–51, 2016.

[41] X. Xiong, L. Yu, W. Yang et al., "A high-throughput stereo-imaging system for quantifying rape leaf traits during the seedling stage," *Plant Methods*, vol. 13, no. 1, pp. 1–17, 2017.

[42] M. Klodt and D. Cremers, "High-resolution plant shape measurements from multi-view stereo reconstruction," European Conference on Computer Vision. Springer, Cham, 2015.

[43] J. C. Rose, S. Paulus, and H. Kuhlmann, "Accuracy analysis of a multi-view stereo approach for phenotyping of tomato plants at the organ level," *Sensors*, vol. 15, no. 5, pp. 9651–9665, 2015.

[44] T. Miao, W. Wen, Y. Li, S. Wu, C. Zhu, and X. Guo, "Label3D-Maize: toolkit for 3D point cloud data annotation of maize shoots," *GigaScience*, vol. 10, no. 5, 2021.

[45] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp, "A novel mesh processing based technique for 3D plant analysis," *BMC Plant Biology*, vol. 12, no. 1, p. 63, 2012.

[46] T. Duan, S. Chapman, E. Holland, G. Rebetzke, Y. Guo, and B. Zheng, "Dynamic quantification of canopy structure to characterize early plant vigour in wheat genotypes," *Journal of Experimental Botany*, vol. 67, no. 15, pp. 4523–4534, 2016.

[47] K. Itakura and F. Hosoi, "Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3D plant images," *Sensors*, vol. 18, no. 10, p. 3576, 2018.

[48] W. Su, D. Zhu, J. Huang, and H. Guo, "Estimation of the vertical leaf area profile of corn (*Zea mays*) plants using terrestrial laser scanning (TLS)," *Computers and Electronics in Agriculture*, vol. 150, pp. 5–13, 2018.

[49] S. Li, L. Dai, H. Wang, Y. Wang, Z. He, and S. Lin, "Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model," *Remote Sensing*, vol. 9, no. 11, p. 1202, 2017.

[50] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 501–508, Zurich, Switzerland, 2012.

[51] D. Zermas, V. Morellas, D. Mulla, and N. Papanikolopoulos, "Estimating the leaf area index of crops through the evaluation of 3D models," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6155–6162, Vancouver, BC, Canada, 2017.

[52] H. Pan, F. Hetroy-Wheeler, J. Charlaix, and D. Colliaux, "Multi-scale space-time registration of growing plants," in *2021 IEEE International Conference on 3D Vision (3DV)*, London, United Kingdom, 2021.

[53] N. Chebrolu, F. Magistri, T. Läbe, and C. Stachniss, "Registration of spatio-temporal point clouds of plants for phenotyping," *PLoS One*, vol. 16, no. 2, article e0247243, 2021.

[54] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[55] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat, "Snapnet-r: consistent 3d multi-view semantic labeling for robotics," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 669–678, Venice, Italy, 2017.

[56] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, Santiago, Chile, 2015.

[57] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3779–3788, Honolulu, HI, USA, 2017.

[58] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Computers & Graphics*, vol. 71, pp. 189–198, 2018.

[59] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection," *Robotics: Science and Systems*, vol. 1, no. 3, pp. 10–15, 2015.

[60] J. Huang and S. You, "Point cloud labeling using 3d convolutional neural network," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2670–2675, Cancun, 2016.

[61] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "Fpnn: field probing neural networks for 3d data," *Advances in Neural Information Processing Systems*, vol. 29, pp. 307–315, 2016.

[62] Z. Wu, S. Song, A. Khosla et al., "3d shapenets: a deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, 2015.

[63] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, Hamburg, Germany, 2015.

[64] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Honolulu, HI, USA, 2017.

[65] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5099–5108, 2017.

[66] D. Li, G. Shi, Y. Wu, Y. Yang, and M. Zhao, "Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2175–2191, 2020.

[67] T. Masuda, "Leaf area estimation by semantic segmentation of point cloud of tomato plants," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, 2021.

[68] Y. Li, W. Wen, T. Miao et al., "Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning," *Computers and Electronics in Agriculture*, vol. 193, article 106702, 2022.

[69] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: similarity group proposal network for 3d point cloud instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2569–2578, Salt Lake City, UT, USA, 2018.

[70] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567, Salt Lake City, UT, USA, 2018.

[71] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5209–5218, Venice, Italy, 2017.

[72] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29–38, Honolulu, HI, USA, 2017.

[73] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4548–4557, Salt Lake City, UT, USA, 2018.

[74] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[75] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4096–4105, Long Beach, CA, USA, 2019.

[76] L. Zhao and W. Tao, "Jsnet: joint instance and semantic segmentation of 3d point clouds," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12951–12958, 2020.

[77] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, Las Vegas, NV, USA, 2016.

[78] F. Zhang, J. Fang, B. Wah, and P. Torr, "Deep fusionnet for point cloud semantic segmentation," in *Computer Vision–ECCV 2020*, pp. 644–663, Springer International Publishing, 2020.

[79] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[80] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.

[81] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, Munich, Germany, 2018.

[82] F. Wang, M. Jiang, C. Qian et al., "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, Honolulu, HI, USA, 2017.

[83] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 7354–7363, Long Beach, CA, USA, 2019.

[84] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, UT, USA, 2018.

[85] A. Conn, U. V. Pedmale, J. Chory, and S. Navlakha, "High-resolution laser scanning reveals plant architectures that reflect universal network design principles," *Cell systems*, vol. 5, no. 1, pp. 53–62.e3, 2017.

[86] A. Conn, U. V. Pedmale, J. Chory, C. F. Stevens, and S. Navlakha, "A statistical description of plant shoot architecture," *Current biology*, vol. 27, no. 14, pp. 2078–2088.e3, 2017.

[87] "The Website of Semantic Segmentation Editor," Septemeber, 2019 https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor/.

[88] I. Armeni, O. Sener, A. R. Zamir et al., "3d semantic parsing of large-scale indoor spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1534–1543, Las Vegas, NV, USA, 2016.