

Article

Automatic Branch–Leaf Segmentation and Leaf Phenotypic Parameter Estimation of Pear Trees Based on Three-Dimensional Point Clouds

Haitao Li ^{1,2}, Gengchen Wu ¹, Shutian Tao ³ , Hao Yin ³, Kaijie Qi ³, Shaoling Zhang ³, Wei Guo ⁴ , Seishi Ninomiya ^{1,4,*}  and Yue Mu ^{1,*} 

- ¹ Academy for Advanced Interdisciplinary Studies, Collaborative Innovation Center for Modern Crop Production Co-Sponsored by Province and Ministry, Nanjing Agricultural University, Nanjing 210095, China
² College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210095, China
³ Centre of Pear Engineering Technology Research, State Key Laboratory of Crop Genetics & Germplasm Enhancement and Utilization, Nanjing Agricultural University, Nanjing 210095, China
⁴ Graduate School of Agricultural and Life Sciences, The University of Tokyo, 1-1-1 Midori-cho, Tokyo 188-0002, Japan
* Correspondence: snino@g.ecc.u-tokyo.ac.jp (S.N.); yuemu@njau.edu.cn (Y.M.)

Abstract: The leaf phenotypic traits of plants have a significant impact on the efficiency of canopy photosynthesis. However, traditional methods such as destructive sampling will hinder the continuous monitoring of plant growth, while manual measurements in the field are both time-consuming and laborious. Nondestructive and accurate measurements of leaf phenotypic parameters can be achieved through the use of 3D canopy models and object segmentation techniques. This paper proposed an automatic branch–leaf segmentation pipeline based on lidar point cloud and conducted the automatic measurement of leaf inclination angle, length, width, and area, using pear canopy as an example. Firstly, a three-dimensional model using a lidar point cloud was established using SCENE software. Next, 305 pear tree branches were manually divided into branch points and leaf points, and 45 branch samples were selected as test data. Leaf points were further marked as 572 leaf instances on these test data. The PointNet++ model was used, with 260 point clouds as training input to carry out semantic segmentation of branches and leaves. Using the leaf point clouds in the test dataset as input, a single leaf instance was extracted by means of a mean shift clustering algorithm. Finally, based on the single leaf point cloud, the leaf inclination angle was calculated by plane fitting, while the leaf length, width, and area were calculated by midrib fitting and triangulation. The semantic segmentation model was tested on 45 branches, with a mean $Precision_{sem}$, mean $Recall_{sem}$, mean $F1-score$, and mean Intersection over Union (IoU) of branches and leaves of 0.93, 0.94, 0.93, and 0.88, respectively. For single leaf extraction, the $Precision_{ins}$, $Recall_{ins}$, and mean coverage ($mCoV$) were 0.89, 0.92, and 0.87, respectively. Using the proposed method, the estimated leaf inclination, length, width, and area of pear leaves showed a high correlation with manual measurements, with correlation coefficients of 0.94 (root mean squared error: 4.44°), 0.94 (root mean squared error: 0.43 cm), 0.91 (root mean squared error: 0.39 cm), and 0.93 (root mean squared error: 5.21 cm^2), respectively. These results demonstrate that the method can automatically and accurately measure the phenotypic parameters of pear leaves. This has great significance for monitoring pear tree growth, simulating canopy photosynthesis, and optimizing orchard management.



Citation: Li, H.; Wu, G.; Tao, S.; Yin, H.; Qi, K.; Zhang, S.; Guo, W.; Ninomiya, S.; Mu, Y. Automatic Branch–Leaf Segmentation and Leaf Phenotypic Parameter Estimation of Pear Trees Based on Three-Dimensional Point Clouds. *Sensors* **2023**, *23*, 4572. <https://doi.org/10.3390/s23094572>

Academic Editor: Steve Vanlanduit

Received: 21 March 2023

Revised: 30 April 2023

Accepted: 5 May 2023

Published: 8 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: pear canopy; point cloud segmentation; leaf phenotype

1. Introduction

Leaves are the primary organs for photosynthesis and respiration in plants, especially fruit trees, and account for the largest proportion in the tree crown. They play a key role in the growth and development of plants, making their characteristics important

for characterizing plant growth conditions [1]. Thus, automatic extraction of phenotypic parameters from leaves is essential for monitoring fruit tree growth [2].

Traditional methods for extracting leaf parameters often require manual measurement, which are time-consuming, laborious, and destructive [3]. Devices based on radiation transmittance measurement have been used for canopy porosity studies, but the sensors could only be put at sample points or line transect, which still takes a lot of sampling time to analyze the complete 3D canopy [4]. Although image-based methods are low-cost and fast, the extraction of leaf phenotypes may be limited in complex scenarios with severe canopy overlap [5,6]. Fortunately, the use of ToF (Time of Flight) camera [7] and lidar in agriculture and forestry allows for the quick and accurate acquisition of three-dimensional information about the canopy, making the extraction of leaf phenotypic parameters as well as plants' (trees') volume efficient and accurate [8–11]. However, extracting leaves from canopy point cloud models to realize single leaf measurement is challenging [12]. The common strategy is to separate branches and leaves first, then extract single leaves, and finally measure leaf phenotypic parameters.

In recent years, branch–leaf segmentation based on point clouds are continuously developing, from using the difference in laser reflection intensity based on branches and leaves, to the use of difference in their spatial structure characteristics, and further obtaining more features through deep learning methods. Cote et al. [13] attempted to classify evergreen conifer trunks and leaves using their different infrared spectral responses. However, the separation of branches and leaves by use of single intensity information is empirical. Xu et al. [14] calculated the shortest path from each point to the root point of the tree, and then used a threshold to distinguish the point cloud of the branches and the leaves. However, the accuracy of this segmentation method is not high enough. Su et al. [15] proposed an extraction algorithm that combines classification and segmentation based on K-means clustering and random sample consensus algorithm (RANSAC) to divide canopy point clouds into branches and leaves. Tang et al. [16] separated the branches and leaves by using both the similarity of principal direction between neighboring points and distribution density of points. Hu et al. [17] used the SegNet network to separate branches and leaves in the depth images and then extract leaf point clouds. Compared with using the discrepancy between reflection intensity and simple spatial structure characteristics of branch and leaf, it is more accurate to segment them through a deep learning model.

In terms of single leaf segmentation based on point clouds, it has been conducted in various plants to trees. For the leaf segmentation of plants, Xia et al. [18] used mean-shift clustering to segment objects from the background and active contour models, while calculating gradient vector field to segment leaves in situ in a greenhouse. Li et al. [19] proposed a single leaf segmentation method based on three-dimensional filtering and facet region growth, which can better segment overlapping leaves. In the leaf segmentation of crops, Duan et al. [20] used the octree algorithm to segment wheat point clouds by single leaf, and estimated phenotypic parameters. Jin et al. [21] proposed the median normalized vector growth algorithm to segment maize stems and leaves based on lidar point clouds through four steps: pre-treatment, stem growth, leaf growth, and post-treatment. In the leaf segmentation of (fruit) trees, Liu et al. [22] used the dynamic K threshold to segment single leaves on the branches of apple trees. However, the point cloud of the branches was not removed before single leaf segmentation, leaving some noise in the point clouds of leaves. Similarly, Wu et al. [23] acquired the point clouds of apple and orange canopies, and adopted Affinity Propagation algorithm to realize the separation of leaves. Koma et al. [24] extracted leaves by region growing for the lidar point clouds of a tulip tree. In addition, the methodology of segmenting poplar single leaves in the literature [17] was also the point cloud clustering method, based on k-d tree. According to the above, we found that methods based on region growing are the most common methods for single leaf segmentation of plants and crops. However, clustering methods based on Euclidean distance are more commonly used in single leaf segmentation of (fruit) trees than plants

and crops, which may be due to the larger canopy, relatively smaller leaves, and scattered leaf distribution.

In addition to the above step-by-step extraction strategy, the application of deep learning methods makes it possible to simultaneously conduct branch–leaf semantic segmentation and segmentation of single leaf instances. Jin et al. [12] proposed a voxel-based convolutional neural network (VCNN) for the stem–leaf semantic segmentation and instance segmentation of maize from terrestrial lidar data. In a similar deep learning approach, Li et al. [25] proposed the PSegNet neural network to segment plant point clouds and trained multi-period tobacco, tomato, and sorghum point clouds using the Voxelized Farthest Point Sampling (VFPS) strategy. Li et al. [26] developed DeepSeg3Dmaize, a plant point cloud segmentation technique that integrates high-throughput data acquisition and deep learning, using PointNet to implement stem–leaf and organ instance segmentation. Compared with the previous two plant/crop models, PointNet is more widely applicable.

Based on single leaf point cloud, phenotypic parameters such as leaf length and width, perimeter, area, leaf inclination, and azimuth angle can be easily estimated. Leaf length and width are the most commonly measured parameters, which can be estimated using the distance between two points on the tip and base [27], or through convex hull [24] and bounding box methods [28,29]. However, due to leaf curvature, these methods may lead to large errors. Using the midrib of the leaf to simplify the point cloud can improve the accuracy [20]. For leaf inclination angle, it can be estimated by the angle between the ventral normal of the leaf and the zenith axis, and the least squares (LS) method are widely used for plane (line) fitting to obtain the normal (directional) vector [26,30]. As for leaf area, it can be estimated with point cloud triangulation and surface reconstruction [19,31,32].

Compared with crops and plants, due to the interlacing branches in the canopy, it is difficult to obtain high-quality canopy point cloud models of fruit trees. In addition, the leaf segmentation methods using deep learning largely rely on datasets, and the annotation of branch/leaf point cloud models of fruit trees is also challenging. Therefore, there are few studies on organ scale (branch and leaf) segmentation for fruit trees based on deep learning, much less based on leaf phenotypic characteristics' measurement, at present. In this study, we aim to develop an automatic pipeline for single leaf extraction and measurement of fruit trees, not only efficient in point cloud processing for fruit tree canopy, but also accurate in measuring leaf phenotypic parameters.

2. Materials and Methods

This section is comprised of five main parts: point cloud data acquisition and preprocessing, construction of the branch level point cloud dataset, branch and leaf point cloud segmentation using deep learning, leaf point cloud segmentation using clustering, and estimation of leaf phenotypic parameters (as shown in Figure 1). The related algorithms and samples are available at https://github.com/haitao971028/branch-leaf_segmentation_and_leaf_traits_extraction (accessed on 1 March 2021).

2.1. Data Acquisition

Pear trees (*Pyrus pyrifolia* 'Cuiguan') with "Y" shape and 7 years of age were taken as materials, planted in Baima Scientific Research Base of Nanjing Agricultural University, Lishui District, Nanjing, Jiangsu Province, China.

In order to obtain accurate point cloud data of pear trees, the three-dimensional laser scanner FARO Focus^{3D} S70 (FARO Technologies, Inc., Lake Mary, FL, USA) was used for point cloud data acquisition. The scanning was conducted in late June 2022 under clear weather conditions, with no wind or light breeze. Due to the severe obscuration between the leaves and branches of the pear tree, the scanner was placed on a tripod with a height of approximately 1.6 m for multi-site scanning, in order to obtain all-round point cloud data. The field of view was 360° horizontal × 300° vertical, and the scanning distance was approximately 5 m (as shown in Table 1). The multi-site cloud registration was completed using FARO SCENE software (FARO Technologies, Inc., Lake Mary, FL, USA), and a total

of 20 pear trees were scanned. To conduct subsequent quantitative evaluation on the estimation of phenotypic parameters, 5 branch samples were randomly selected from the pear trees for indoor scanning (as shown in Figure 1a), with the scanning parameters set the same as those in the field. Additionally, a total of 50 leaves were selected from the 5 branch samples, and the corresponding leaf dip angle, length, width, and area were manually measured as the true values.

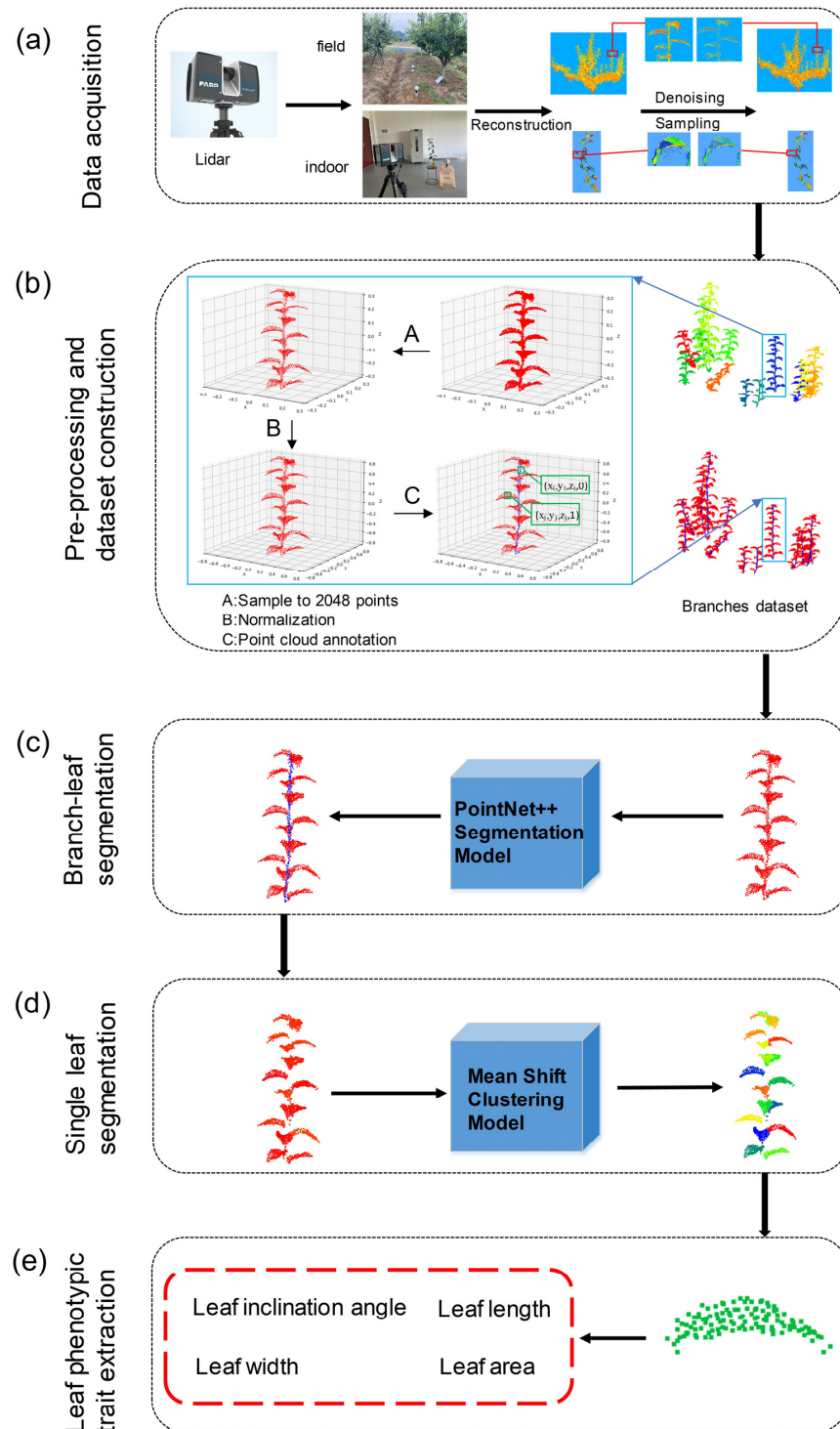


Figure 1. Workflow of this study: (a) Data acquisition; (b) Data pre-processing and branch level dataset construction; (c) Branch-leaf segmentation with PointNet++ segmentation model; (d) Single leaf segmentation with Mean Shift Clustering Model; (e) Leaf phenotypic trait extraction.

Table 1. Specifications of lidar scanners used in this study.

Index	FARO Focus ^{3D} S70
Laser Class	Level 1 (IEC 60825-1)
Laser wavelength	1550 nm
Detection range	0.6–70 m
Field of view	horizontal 360° × vertical 300°
Single point measuring speed	Up to 976,000 times/s
Scanner weight	4.2 kg
Sensors	Inclinometer, compass, GPS, height sensor, dual axis compensator
Scanning point spacing (scanning distance)	0.003 m (10 m)
Measuring error in distance	0.001 m

2.2. Dataset Construction

2.2.1. Data Preprocessing

Using the Cloud Compare V2 (CC) point cloud visualization software (<http://www.cloudcompare.org/>, accessed on 1 March 2021) and the Point Cloud Library (PCL, Kitware Inc., Tallahassee, FL, USA), the denoising and thinning of the pear tree point clouds were completed [33]. The specific processes were as follows: Firstly, the ground and trunk point clouds were manually removed using CC to obtain the whole canopy point clouds. Then, Statistical Outlier Removal algorithm in the PCL was used to remove outliers. Finally, the point cloud was voxelized to implement thinning, with the length, width, and height of the voxels set to 0.003 m.

We used a semi-automated method to extract branches from the canopy. Firstly, CC software was used to select some relatively complete clusters of branches from the whole canopy for preliminary extraction. Then, cluster segmentation of branches was performed based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [34] and K-Nearest Neighbor (KNN) [35] algorithm to obtain single branches. For a few serious overlapping branches, we manually segmented them by CC software. A total of 373 branch samples, including 5 samples scanned indoors, were taken from the 20 pear canopies point cloud models to establish the branch dataset. The number of points for each branch ranged from 3 k to 10 k. To improve model training efficiency and retain the structural characteristics of branches, the Farthest Point Sampling (FPS) method was used to sample the branch point cloud to 2048 points. The point cloud was then normalized with the origin as the center into a cube with a side length of 2 m. To avoid affecting subsequent parameter measurement, the conversion parameter from the original point cloud model to the normalized output was recorded, and was later used to reverse the normalized point cloud to original scale and measure its real size.

2.2.2. Point Cloud Labeling

After data preprocessing, point cloud annotation is necessary to implement the following model training. In this study, labels were set at the point level. Since the data samples consisted only of branches and leaves, we assigned a label of “1” to point clouds belonging to leaves and “0” to branch point clouds (as shown in Figure 1b). The labeling was done using CC software.

The format of point cloud data in this study is an $n \times 4$ matrix, where n is the number of points in the sample. The matrix consists of four columns: the x , y , and z coordinates of the points, and a label column with values of 0 or 1.

2.2.3. Dataset Partitioning

In order to ensure the quality of the dataset, 373 branches were selected by considering the diversity of leaf density, leaf distribution, and the completeness of the branch and leaves. Finally, we obtained a total of 305 high-quality branch samples (including 5 indoor samples). Additionally, we conducted statistics on this dataset. The length of all branches ranged from 0.25 m to 1.13 m, and the number of leaves on each branch ranged from

8 to 26, as shown in Table 2. Out of the 305 samples, 260 samples (all infield) were randomly selected for training, and another 45 (40 infield and 5 indoor) were used as test samples.

Table 2. Statistics of branch length and leaf number of samples in training set and test set.

Trait	Training Set				Test Set			
	Maximum Value	Minimum Value	Mean Value	Standard Deviation	Maximum Value	Minimum Value	Mean Value	Standard Deviation
Branch length (m)	1.06	0.25	0.57	0.16	1.13	0.36	0.58	0.17
Leaf number	26	8	14.6	3.6	25	8	14.3	3.7

2.3. Branch–Leaf Segmentation Based on PointNet++ Model

2.3.1. PointNet++ Segmentation Model

PointNet++ [36] is a deep neural network capable of directly processing disordered point cloud data. It is an upgraded version of PointNet [37] that addresses the limitations of the PointNet network with regard to local feature extraction. The network is primarily used for point cloud classification and segmentation. In this experiment, the segmentation network of PointNet++ is utilized for branch–leaf segmentation.

The segmentation network is comprised of an encoder and a decoder. The encoder is primarily responsible for the point cloud downsampling process, and extracting the local features of the point cloud by setting up multiple Set Abstraction structures. The Set Abstraction is composed of sampling, grouping, and PointNet modules, which eventually output a point cloud with global features. The decoder, on the other hand, is responsible for the upsampling process. The downsampled points are restored through distance-based interpolation, and the characteristics of each point are calculated based on the KNN, which are then sent to Softmax to achieve point-level classification. Figure 2 shows the structure of the segmentation network.

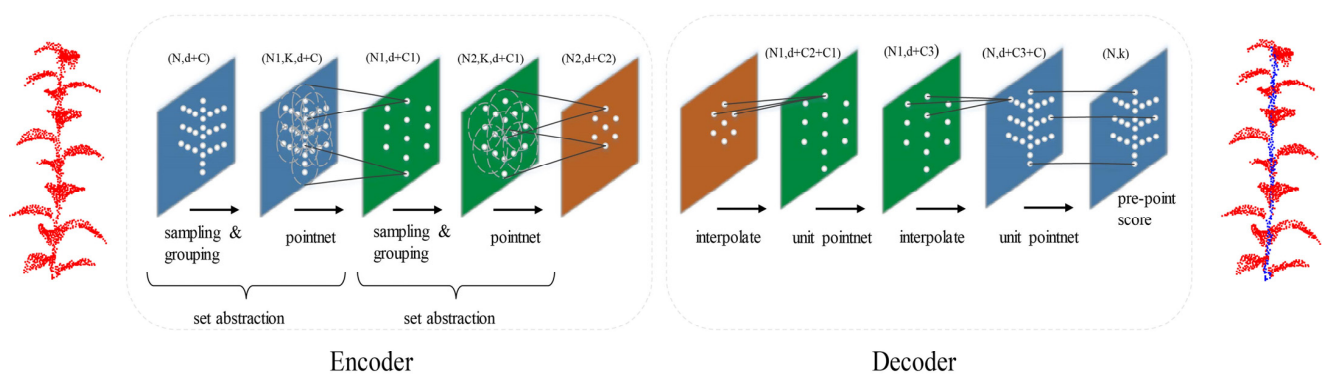


Figure 2. Structure of PointNet++ segmentation network. N represents the number of points, K represents the number of groups, d represents the coordinate dimension, and C represents the feature dimension.

2.3.2. Model Training

The PointNet++ model was trained using the PyTorch (<https://pytorch.org/>) framework. The training set was input the network with a batch size of 4. The initial learning rate was set to 0.001 and dynamically adjusted using the ADAM optimizer and stochastic gradient descent (SGD). The momentum was set to 0.9, and the weight attenuation coefficient was set to 0.001.

In this experiment, the PointNet++ segmentation network was iterated for 500 epochs. The network was trained on an Ubuntu 16.04 OS, with an Intel Xeon E5-2698V4 CPU, 256 GB of memory, and NVIDIA Tesla V100 GPU.

2.4. Single Leaf Segmentation Based on Mean Shift Clustering

PointNet++ performed semantic segmentation of the points of branches and leaves at branch level, but did not segment single leaves. Therefore, we then utilized the coordinate information of the point cloud to conduct mean shift clustering [38] in three-dimensional space for instance segmentation of single leaves. Unlike other clustering algorithms, the mean shift algorithm is based on centroids. It can identify the dense center of data points by radius and cluster according to the density center without specifying a number of clusters [39].

The algorithm requires setting the key parameter radius. To ensure that the clustering center is closer to the leaf centroid, the size of the radius was set as the radius of the circumscribed sphere of the leaf point cloud. In this study, three different radius sizes of 35 mm, 45 mm, and 55 mm were set, based on the actual size of pear tree leaves.

2.5. Estimation of Phenotypic Parameters

Based on the single leaf point cloud, four phenotypic parameters, including leaf inclination angle, leaf length, leaf width, and leaf area, were estimated, as shown in Figures 1e and 3. When using multi-station lidar scan for registering, the leaf surface in point cloud may not be smooth due to wind and registration errors, which has a significant impact on leaf surface reconstruction in the later stage. Therefore, the Moving Least Squares (MLS) method [40] was employed to resample the point clouds (i.e., smoothing) before parameter estimation, as shown in Figure 3b.

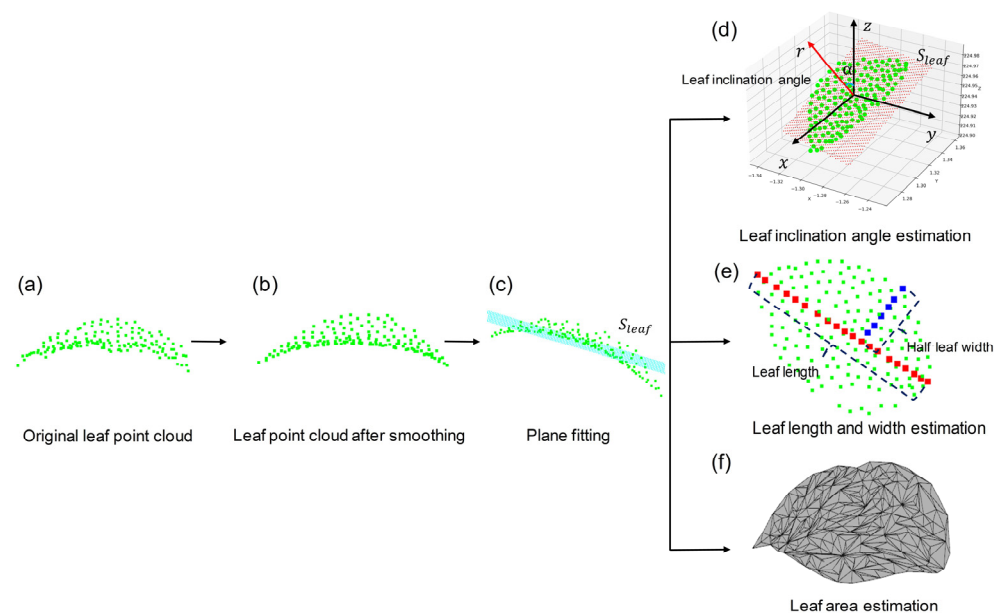


Figure 3. Schematic diagram of phenotypic parameters estimation based on single leaf point cloud. (a) Original leaf point cloud; (b) Leaf point cloud after smoothing using MLS; (c) Leaf point cloud plane fitting; (d) Leaf inclination angle estimation; (e) Leaf length and width estimation; (f) Leaf area estimation. S_{leaf} in (c,d) is the leaf point cloud fitting plane.

2.5.1. Estimation of Leaf Inclination Angle

The leaf inclination angle is the angle between the ventral normal γ of the leaf and the zenith axis z , ranging from 0° to 90° . In this study, the normal vector r of leaf fitting plane S_{leaf} was used to approximate the normal vector of the leaf point cloud, and the leaf inclination α was the angle between r and z , as shown in Figure 3d. To obtain S_{leaf} and r , the least squares method was used [30].

2.5.2. Estimation of Leaf Length and Width

In order to improve the accuracy of parameter estimation, this study proposed the midrib fitting algorithm to extract leaf length and width from point cloud, as shown in Figure 4.

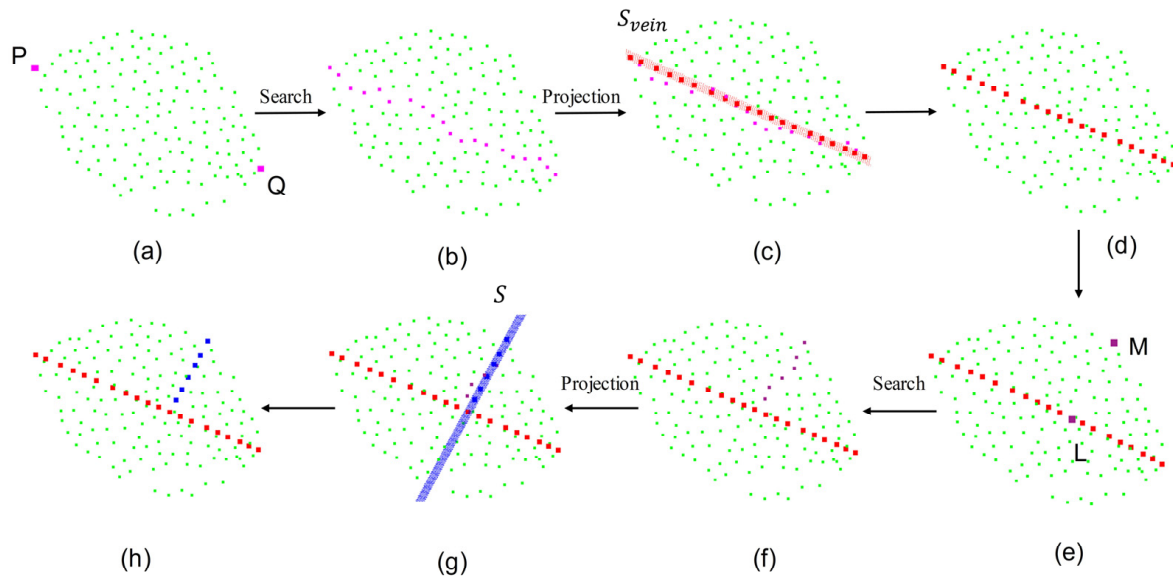


Figure 4. Schematic diagram of estimating leaf length and width based on midrib fitting. (a) The leaf base point P and the leaf tip point Q; (b) Approximate point cloud (purple points) of midrib; (c) The approximate midrib point cloud is projected onto the plane S_{vein} ; (d) Midrib fitting point cloud (red points) after projection; (e) The starting point M and the ending point L when estimating the leaf width; (f) Approximate point cloud (dark purple points) of estimating leaf width; (g) The approximate point cloud of estimating leaf width is projected onto the plane S; (h) Fitting point cloud for estimating leaf length (red points) and leaf width (blue points). S_{vein} in (c) is the projection plane of midrib point cloud, and S in (g) is the widest cross section of the leaf.

Based on the morphology of pear tree leaves and the leaf point cloud, it can be estimated that the two points with the furthest distance from the leaf point cloud are the leaf base point P and the leaf tip point Q (Figure 4a). Using these two points, the K-nearest neighbor algorithm was employed to approximate the midrib. The algorithm involves the following steps:

1. Set the two points obtained above as the starting and ending points, respectively (no need to specify which is the starting point);
2. Take the starting point as the leaf base point and add it to the base point set;
3. Establish a k-d tree [41] of the leaf point cloud and search for the 1st to Kth nearest neighbor points of the base point;
4. Calculate the distance 1 between one of these neighbor points and the base point, and the distance 2 between it and the ending point. Sum up distance 1 and distance 2 and denote this as D. Repeat this until the D of all the points is calculated;
5. Find the point N that minimizes D;
6. Add N to the base point set, set N as the new base point, and remove N from the leaf point cloud;
7. Repeat steps 3, 4, 5, 6 until the base point equals the endpoint;
8. Collect the base points.

The pseudo code of the midrib fitting algorithm is as follows (Algorithm 1):

Algorithm 1 Midrib fitting algorithm**Inputs:** Point cloud I **Parameters:** Starting point s , endpoint e , base point $b = s$, and k .**Outputs:** Point cloud O after midrib fittingDefine three local variables N, D and d $O \leftarrow \emptyset, N \leftarrow (0,0,0), D \leftarrow 0$, and $d \leftarrow +\infty$ $O.push_back(b)$.**while** $b \neq e$ **do**Establish the k -d tree of O .Initialize b 's k nearest neighbors n_k **for each** n , in n_k **do**Compute D , the sum of the distance between n and b , and the distance between n and e .**if** $D < d$ **then** $N \leftarrow n$ $d \leftarrow D$ **end if****end for** $O.push_back(N)$, and $I.erase(N)$. $b \leftarrow N$ **end while**

After the above steps, a point cloud approximation of the midrib can be obtained (Figure 4b). To improve the fitting of the midrib, the above point cloud is projected onto the plane S_{vein} , which passes through the points P and Q and is perpendicular to S_{leaf} , to obtain a new point cloud of the midrib (Figure 4d). The distance between adjacent points is calculated, and the leaf length can be approximated by adding them up.

The calculation method for leaf width is similar to that of leaf length. First, the starting point M is obtained, which is a boundary point at the widest cross section of the leaf, by finding the farthest distance from the leaf point cloud to S_{vein} (Figure 4e). Then, the distance between point M and each point in the fitted midrib point cloud is calculated. If the distance between points L and M is the shortest, L is the desired end point (Figure 4e). Finally, based on L and M, the midrib fitting algorithm can be used to acquire the point for calculating leaf width (Figure 4f). In this case, the projection plane is S (Figure 4g), which passes through point M and is perpendicular to S_{leaf} and S_{vein} . The distance between adjacent points is calculated, and half of the leaf width can be approximated by summing these distances.

2.5.3. Estimation of Leaf Area

Greedy Projection Triangulation algorithm [32] is adopted to build the mesh of single leaf point cloud, as shown in Figure 3f.

Using Helen's formula to calculate the area of each triangle and sum up to approximate the leaf area $Area_{leaf}$, the formula is as follows:

$$Area_{leaf} = \sum_{i=1}^n \sqrt{p_i(p_i - a_i)(p_i - b_i)(p_i - c_i)} \quad (1)$$

$$p_i = \frac{1}{2}(a_i + b_i + c_i) \quad (2)$$

where a_i, b_i, c_i and p_i are the three sides and half of the perimeter of the i th triangle, respectively.

2.6. Evaluation Metric

In this study, we employed various evaluation methods for the branch-leaf and single leaf segmentation. To evaluate the performance of the semantic segmentation of branches and leaves, we introduced four metrics: $Precision_{sem}$, $Recall_{sem}$, $F1-score$, and Intersection

over Union (*IoU*) [42]. For each semantic class, the *IoU* reflects the degree of overlapping between the predicted results of each semantic category and the corresponding real results. $Precision_{sem}$ reflects the proportion of points that the network correctly predicted in the total number of points predicted in the corresponding category. $Recall_{sem}$ refers to the ratio of the number of points the network correctly predicted to the total number of points in this category. *F1-score* is the harmonic mean of $Precision_{sem}$ and $Recall_{sem}$, and its value ranges between 0 and 1. Higher values for these four indicators indicate better segmentation performance. The four metrics are defined as follows:

$$Precision_{sem} = \frac{TP}{TP + FN} \quad (3)$$

$$Recall_{sem} = \frac{TP}{TP + FP} \quad (4)$$

$$F1\text{-score} = 2 \times \frac{Precision_{sem} \times Recall_{sem}}{Precision_{sem} + Recall_{sem}} \quad (5)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (6)$$

Among them, *TP* refers to a point that is correctly predicted in this class, i.e., belongs to the same class as manually labeled. *FN* refers to a point manually labeled in this class but is incorrectly predicted to be in another class. *FP* refers to a point that is not manually labeled in this class but is predicted to belong to it. These metrics were calculated for both branches and leaves, and the averages were used for comprehensive evaluation.

For the evaluation of single leaf segmentation, the mean coverage (*mCov*) was used [43]. *mCov* represents the average point-level *IoU* matching between predicted and manually marked instance, which is defined as follows:

$$mCov(I, P) = \frac{1}{I} \sum_{m=1}^I \max_n (IoU(I_m, P_n)) \quad (7)$$

where *I* represents the number of all instances, I_m represents the real point set of the *m*th instance, and P_n represents the predicted point set of the *n*th instance. The calculation of *IoU* is the same as that in semantic segmentation.

In addition to point-level evaluation, all instances with *IoU* higher than 0.5 were counted and evaluated at the instance level using two metrics: $Precision_{ins}$ and $Recall_{ins}$. The definitions are as follows:

$$Precision_{ins} = \frac{T}{P} \quad (8)$$

$$Recall_{ins} = \frac{T}{G} \quad (9)$$

where *T* is the number of *IoUs* greater than 0.5 in predicted instances and manually marked instances, *P* is the total number of predicted instances, and *G* is the number of manually marked instances.

The measurements of each phenotypic parameter were evaluated by correlation analysis. The error and accuracy of each parameter were quantified by calculating Root Mean Square Error (*RMSE*) and determination coefficient R^2 between the estimated values and the manually measured values.

3. Results

3.1. Branch–Leaf Segmentation

The point clouds of 45 branch samples in the test dataset were segmented into branches and leaves, and the results were visually and quantitatively evaluated. Figure 5 shows

the results of semantic segmentation of branches and leaves with different attributes, i.e., branch length and number of leaves, and the mean *IoU* of branch and leaf segmentation.

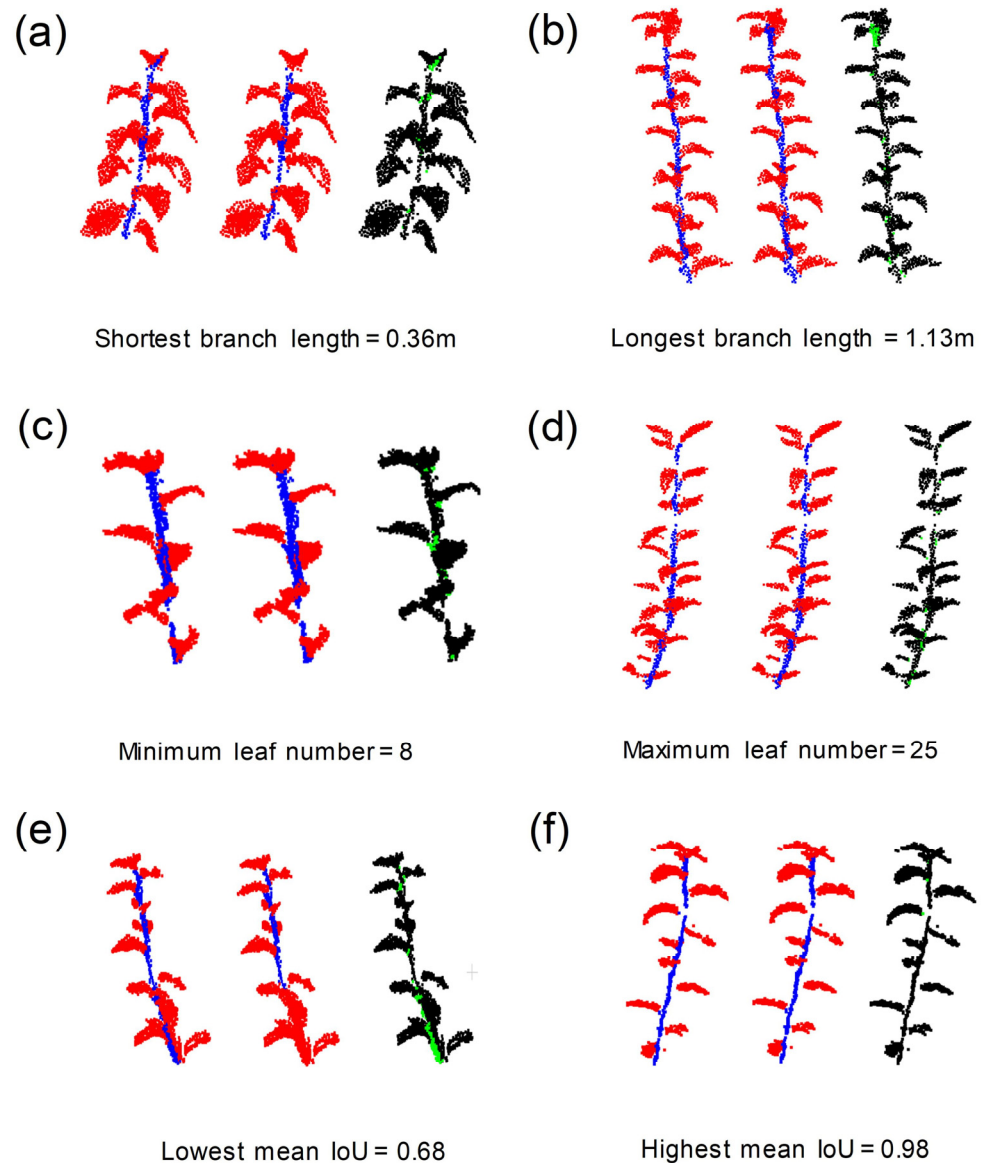


Figure 5. Visualization of branch–leaf semantic segmentation of branches with different attributes in the test dataset using PointNet++. In each subgraph, the left side shows the manual labeling, the middle shows the model prediction (branch and leaf points are in blue and red, respectively), and the right side shows the difference between them (same and different points of classification are in black and green, respectively).

By observing the results of branch–leaf segmentation, we found that the output predicted by the model was very close to manual labeling. However, there were some subtle differences in the junction of branches and leaves, especially at regions near the top of the branch where the leaf clusters sheltered the branches. For instance, some leaves at the top of a branch (Figure 5b) were misclassified as branches, while the branches attached to the leaves (Figure 5e) were misclassified as leaves.

The quantitative evaluation results are displayed in the Table 3. The mean $Precision_{sem}$, mean $Recall_{sem}$, mean $F1-score$, and mean IoU of the semantic segmentation of branches and leaves are 0.92 (Max: 0.99, Min: 0.73), 0.95 (Max: 0.99, Min: 0.85), 0.93 (Max: 0.99, Min: 0.79), and 0.88 (Max: 0.98, Min: 0.68), respectively. There are no significant differences in the

results of infield scanning and indoor scanning for branch–leaf segmentation. Additionally, the segmentation results of this method for samples of different branch lengths and numbers of leaves showed little difference, demonstrating good robustness (as shown in Figure 6).

Table 3. Evaluation of semantic segmentation of branches and leaves of samples in the test dataset.

Trait	Mean $Precision_{sem}$			Mean $Recall_{sem}$			Mean $F1-Score$			Mean IoU		
	Infield	Indoor	All	Infield	Indoor	All	Infield	Indoor	All	Infield	Indoor	All
Maximum	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.98
Minimum	0.46	0.85	0.73	0.58	0.78	0.85	0.62	0.81	0.79	0.45	0.71	0.68
Mean	0.92	0.94	0.92	0.95	0.93	0.95	0.93	0.93	0.93	0.88	0.89	0.88

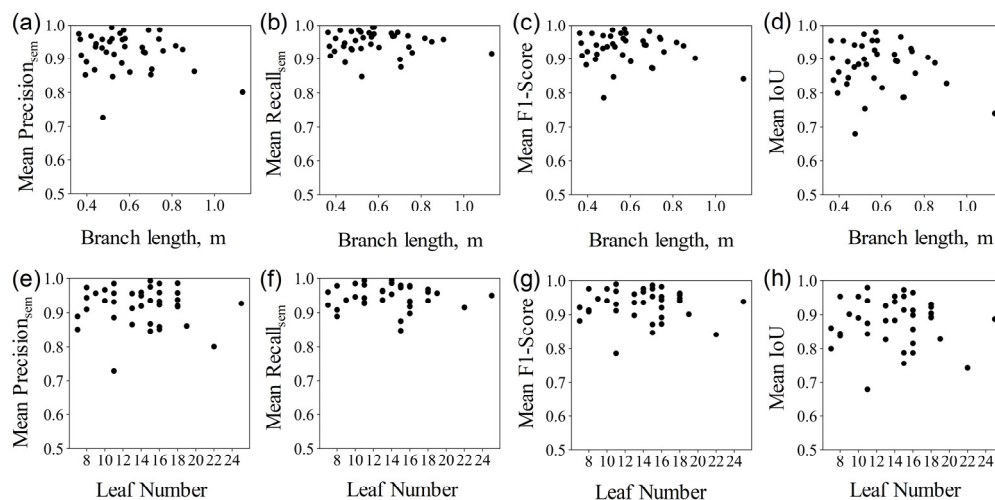


Figure 6. Distribution of the branch–leaf segmentation evaluation metrics with different branch lengths and numbers of leaves. Each subfigure shows mean $Precision_{sem}$, mean $Recall_{sem}$, mean $F1-score$, and mean IoU of each sample with different branch length (subfigures (a–d)) and leaf number (subfigures (e–h)), respectively.

3.2. Single Leaf Segmentation

Based on the results of branch–leaf segmentation, we performed single leaf segmentation for leaf point clouds using mean shift clustering with different radii. From the examples presented in Figure 7, we found that after separating the branches and leaves in the previous step, most leaves could be segmented into single leaves through mean shift clustering. However, some small leaves with missing parts caused over-segmentation. When the radius was too small, one leaf was mistakenly divided into multiple leaves, as shown in Figure 7a. Conversely, when the radius was too large, two or more leaves were grouped into a single leaf. This problem was more obvious where there were leaves at the top of the branch, which were very small and close to each other, as shown in Figure 7c. We found that setting the radius to 45 mm (as shown in Figure 7b) achieved a more balanced segmentation effect.

To evaluate the segmentation effect, the instance-level precision ($Precision_{ins}$), instance-level recall ($Recall_{ins}$) and mean coverage ($mCov$) with different clustering radii were quantitatively evaluated. The segmentation results corresponding to different radii are shown in Table 4. The results show that, using the radius of 45 mm for mean shift clustering could better segment single leaves of pear tree branches. The single leaf segmentation of branches in the test dataset with different attributes (branch length and number of leaves) and $mCov$ are shown in Figure 8. The results show that leaves can be separated from each other effectively with the given radius. However, some large leaves were mistakenly divided into two, as seen in Figure 8b,e.

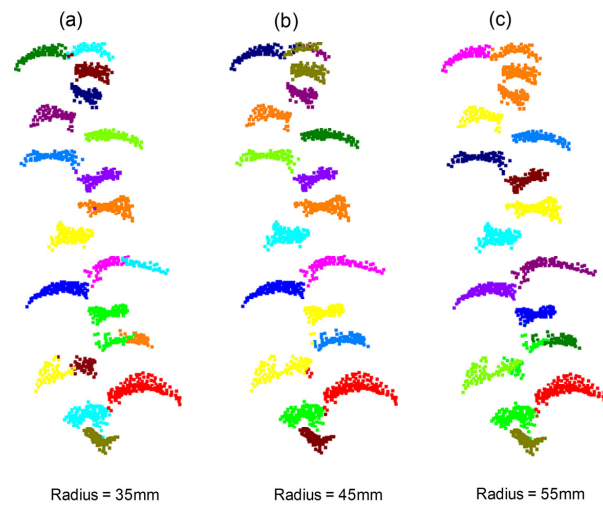


Figure 7. Examples of single leaf segmentation with different radius using mean shift clustering algorithms.

Table 4. Evaluation of single leaf segmentation using mean shift clustering with different radii.

Radius/mm	$Precision_{ins}$	$Recall_{ins}$	$mCov$
35	0.79	0.74	0.76
45	0.89	0.92	0.87
55	0.73	0.82	0.72

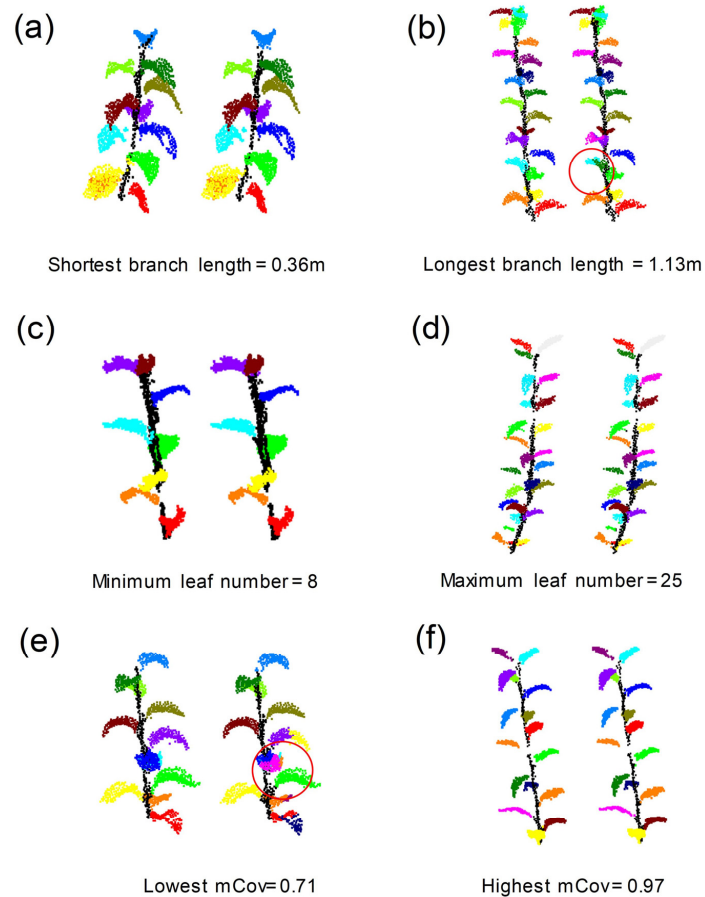


Figure 8. Visualization of single leaf segmentation of branches with different attributes using mean shift clustering algorithm (radius: 45 mm). In each subgraph, the left and right sides are the result of manual and automatic segmentation, respectively. Different leaves are represented by different colors.

The quantitative evaluation results presented in Table 5 show that the $Precision_{ins}$, $Recall_{ins}$, and $mCov$ of single leaf segmentation were 0.89 (Max:0.95, Min:0.68), 0.92 (Max:0.98, Min:0.74), and 0.87 (Max:0.97, Min:0.71), respectively. As well, the results of samples scanned indoors are better than infield for single leaf segmentation. For most samples, this method also showed good robustness of single leaf segmentation (as shown in Figure 9).

Table 5. Evaluation of single leaf segmentation infield and indoors.

	$Precision_{ins}$			$Recall_{ins}$			$mCov$		
	Infield	Indoor	All	Infield	Indoor	All	Infield	Indoor	All
Maximum	0.96	0.98	0.95	0.98	0.99	0.98	0.96	0.98	0.97
Minimum	0.65	0.87	0.68	0.72	0.89	0.74	0.70	0.82	0.71
Mean	0.89	0.93	0.89	0.92	0.95	0.92	0.87	0.94	0.87

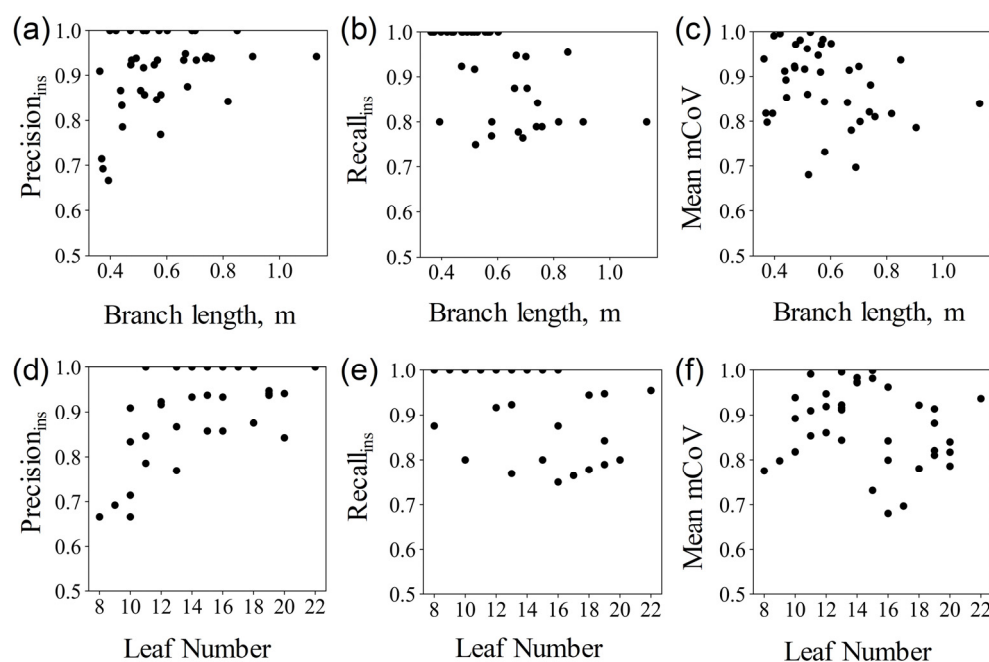


Figure 9. Distribution of the single leaf segmentation evaluation metrics with branch length and number of leaves. Each subfigure shows $Precision_{ins}$, $Recall_{ins}$, and mean $mCov$ of each sample with different branch length (subfigures (a–c)) and leaf number (subfigures (d–f)), respectively.

3.3. Estimation of Phenotypic Parameters

Based on the single leaf point cloud, the leaf inclination angle was calculated by plane fitting, and the leaf length, width, and area parameters were calculated by midrib fitting and triangulation. The estimated phenotypic parameters were compared with manual measurements to evaluate their accuracy by correlation analysis, as shown in Figure 10. The results show a high correlation between the estimated values and the measured values. For leaf inclination angle, the R^2 and $RMSE$ were 0.94 and 4.44° , respectively. The R^2 and $RMSE$ of leaf length were 0.94 and 0.43 cm, respectively, while the R^2 and $RMSE$ of leaf width were 0.91 and 0.39 cm, respectively. In addition, the R^2 and $RMSE$ of leaf area were 0.93 and 5.21 cm^2 , respectively. The midrib fitting method proposed in this study has obtained higher accuracy in estimating leaf length than in width, which may be due to leaf width estimation being more sensitive to curled leaves. In addition, the estimation of leaf area was slightly underestimated. In order to reduce the amount of calculation, we downsampled the original point cloud without changing its structure, which had no effect on the estimation of other phenotypic parameters except the leaf area. As shown in

Figure 10d, the discrepancy between estimated and measured leaf area may be caused by the loss of boundary points of leaf point cloud after downsampling.

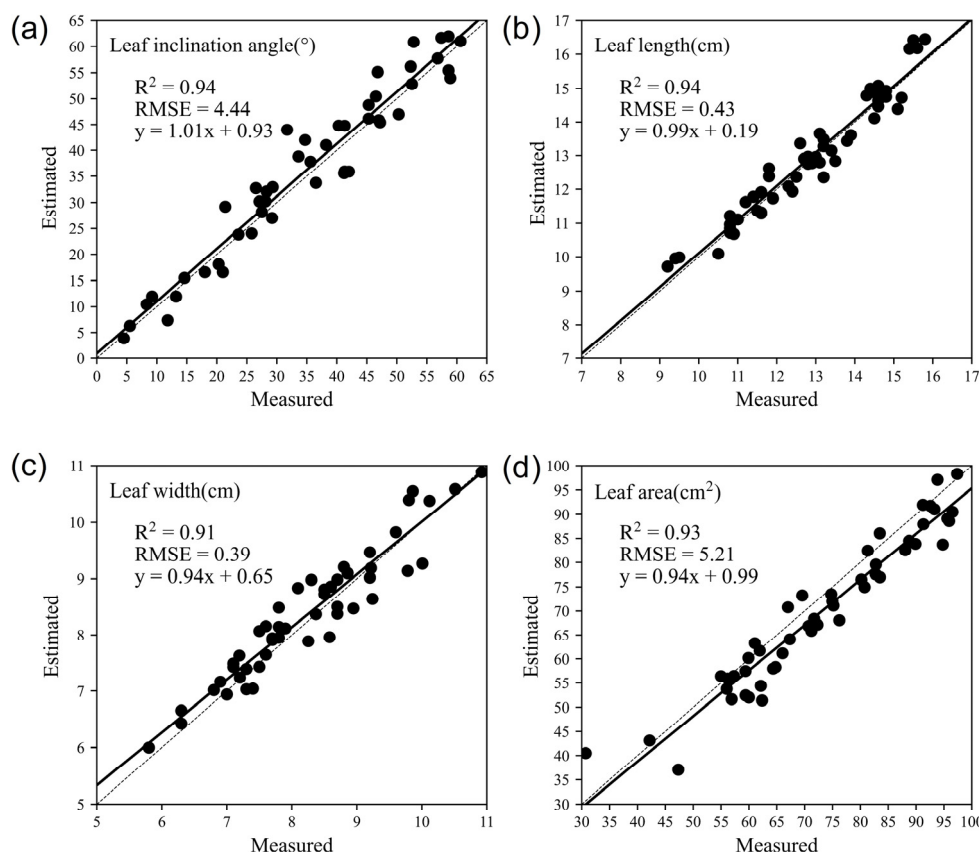


Figure 10. Comparison of phenotypic parameters estimated by the proposed method with the measured values: (a) leaf inclination angle; (b) leaf length; (c) leaf width; and (d) leaf area.

4. Discussion

4.1. Comparison of Similar Studies

The branch–leaf segmentation by PointNet++ achieved relatively high accuracy for pear trees. The mean *IoU* achieving in this study on branch–leaf segmentation reached 0.88, higher than that in literature [15], which used SegNet with Kinect V2 camera, and obtained mean *IoU* of 0.72. In addition to the processing method, terrestrial lidar is more effective to acquire the relatively complete and sophisticated three-dimensional canopy scans of (fruit) trees. However, the mean precision (0.95), recall (0.94), *F1-score* (0.95), and *IoU* (0.90) of branch (stem)–leaf segmentation by Psegnet [25] were slightly higher than our results (precision: 0.93, recall: 0.94, *F1-score* 0.93, and *IoU*: 0.88), owing to its special modules for plants, the double-neighborhood feature extraction block, the double-granularity feature fusion module, and the attention module. In the future, we will further improve accuracy by developing the appropriate modules for fruit trees.

Moreover, the single leaf segmentation by mean shift clustering also obtained acceptable results. In this study, the precision and recall of single leaf segmentation reached 0.89 and 0.92, higher than that in the literature [17], which obtained 0.78 and 0.87 (threshold = 8 mm) by using a geometric distance-based *k*-*d* tree. This is also slightly higher than that by Psegnet [25], which achieved highest precision of 0.90 and *mCov* of 0.85 (ours: 0.87) for tomato leaf, as well as highest recall of 0.82 for sorghum leaf. This may be because Psegnet is designed for segmenting point clouds of several different species of plants, while our approach is specifically for pear trees.

In terms of leaf phenotypic parameter estimation, the correlation coefficients of leaf length, leaf width, leaf inclination angle, and leaf area were 0.94, 0.91, 0.94, and 0.93, respec-

tively. These are a little higher than the results for maize shoot in the literature [26], which were 0.90, 0.82, and 0.94 for leaf length, leaf width, and leaf inclination angle. Additionally, the correlation coefficients of leaf inclination angle and leaf area by our approach exceed those estimated in the literature [31], which were 0.90 and 0.87. Therefore, the phenotypic parameter estimation methods proposed in this study achieved high accuracy, especially for leaf length and width, using a midrib fitting algorithm.

4.2. Limitations

For single leaf segmentation, the proposed method performs better for indoor samples than infield. As shown in Tables 3 and 5, for branch–leaf segmentation, the mean *F1-score* and mean *IoU* of infield and indoor samples were quite similar. At the same time, for single leaf segmentation, the indoor samples outperformed the infield samples for all the three metrics. In the branches' dataset, point clouds in samples were occasionally incomplete due to the overlapping among branches and leaves. In general, due to no wind influence nor overlapping by other branches, the samples collected indoors were of higher quality than infield, mainly in the completeness and accuracy. The difference in data quality between infield and indoor samples mainly affected the single leaf segmentation, since branch and leaf segmentation used deep learning models which were trained with a large number of samples, including incomplete and non-smooth leaves. However, the single leaf segmentation used clustering, and the size of the incomplete leaves was smaller than the average size, so the clustering parameters were not appropriate for their segmentation, resulting in the deviation. In the future, we will integrate the single leaf segmentation with branch–leaf segmentation in the deep learning network to improve its robustness and expand its applications.

In addition, the leaf phenotypic estimation in this paper is more suitable for relatively complete leaves. In order to reduce the impact of branch and leaf occlusion on data acquisition, multi-angle imaging and multi-station lidar scanning are widely used in the construction of three-dimensional plant models. However, due to the serious overlap in fruit trees' canopies, defects in point clouds occur even with multi-station lidar scanning. Incomplete leaves have a significant effect on estimation of phenotypic parameters. For example, if there are missing points in the base or tip of the leaf, the leaf length, width, and area estimated by the proposed midrib fitting algorithm will be underestimated. Currently, point cloud repair is the common approach to solve this problem, and our next plan is to repair the incomplete leaves to further improve the accuracy of estimation for leaf phenotypic parameters.

5. Conclusions

In this paper, we proposed an automatic pipeline for branch–leaf segmentation and leaf phenotypic parameter measurement for pear trees based on lidar point cloud. The method segments branch–leaf point clouds based on the PointNet++ model, extracts single leaf data by mean shift clustering algorithm, and estimates leaf inclination angle, length, width, and area by plane fitting, midrib fitting, and triangulation. It achieved high accuracy in branch–leaf segmentation, single leaf extraction, and leaf phenotypic parameter estimation. For branch–leaf segmentation, the mean *Precision_{sem}*, *Recall_{sem}*, *F1-score*, and *IoU* reached 0.93, 0.94, 0.93, and 0.88, respectively. For single leaf extraction, the *Precision_{ins}*, *Recall_{ins}*, and mean coverage (*mCov*) were 0.89, 0.92, and 0.87, respectively. The correlations between the estimated leaf inclination angle, length, width, and area and manual measurements were 0.94, 0.94, 0.91, and 0.93, respectively. The results demonstrate that the proposed pipeline could efficiently and accurately measure pear leaf phenotypic parameters, which could provide supporting data for monitoring pear tree growth, simulating canopy photosynthesis, and optimizing orchard management.

Author Contributions: Conceptualization, H.L., Y.M. and S.N.; methodology, H.L. and Y.M.; formal analysis, H.L.; investigation, H.L. and G.W.; data curation, H.L.; writing—original draft preparation, H.L. and Y.M.; writing—review and editing, Y.M. and S.N.; visualization, H.L.; supervision, Y.M., W.G. and S.N.; resources, S.T., H.Y., K.Q. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by grants from the National Natural Science Foundation of China (No. 32001980), the Jiangsu Agriculture Science and Technology Innovation Fund (No. CX (21) 3016), and the interdisciplinary project of College of Horticulture (YYJC202201), Nanjing Agricultural University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: This study was made possible by the sponsorship and support of National Natural Science Foundation of China, the Jiangsu Agriculture Science and Technology Innovation Fund, Nanjing Agricultural University, and The University of Tokyo.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CC	Cloud Compare V2
PCL	Point Cloud Library
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
KNN	K-Nearest Neighbor
SGD	Stochastic Gradient Descent
LS	Moving Least Squares
MLS	Least Squares
k-d tree	K-dimensional Tree
IoU	Intersection over Union
mCov	Mean Coverage
RMSE	Root Mean Square Error

References

1. Cristofori, V.; Roupheal, Y.; Gyves, E.M.D.; Bignami, C. A simple model for estimating leaf area of hazelnut from linear measurements. *Sci. Hort.* **2007**, *113*, 221–225. [[CrossRef](#)]
2. Ninomiya, S.; Baret, F.; Cheng, Z.M. Plant Phenomics: Emerging Transdisciplinary Science. *Plant Phenomics* **2019**, *2019*, 2765120. [[CrossRef](#)] [[PubMed](#)]
3. Li, L.; Zhang, Q.; Huang, D. A review of imaging techniques for plant phenotyping. *Sensors* **2014**, *14*, 20078–20111. [[CrossRef](#)] [[PubMed](#)]
4. Castillo-Ruiz, F.J.; Castro-Garcia, S.; Blanco-Roldan, G.L.; Sola-Guirado, R.R.; Gil-Ribes, J.A. Olive Crown Porosity Measurement Based on Radiation Transmittance: An Assessment of Pruning Effect. *Sensors* **2016**, *16*, 723. [[CrossRef](#)] [[PubMed](#)]
5. 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. *Ann. Bot.* **2018**, *121*, 1079–1088. [[CrossRef](#)] [[PubMed](#)]
6. Li, D.W.; Xu, L.H.; Tang, X.S.; Sun, S.Y.; Cai, X.; Zhang, P. 3D Imaging of Greenhouse Plants with an Inexpensive Binocular Stereo Vision System. *Remote Sens.* **2017**, *9*, 508. [[CrossRef](#)]
7. Castillo-Ruiz, F.J.; Colmenero-Martinez, J.T.; Bayano-Tejero, S.; Gonzalez-Sanchez, E.J.; Lara, F.M.; Blanco-Roldán, G.L. Methodology for Olive Pruning Windrow Assessment Using 3D Time-of-Flight Camera. *Agronomy* **2021**, *11*, 1209. [[CrossRef](#)]
8. Garrido, M.; Paraforos, D.S.; Reiser, D.; Vazquez Arellano, M.; Griepentrog, H.W.; Valero, C. 3D Maize Plant Reconstruction Based on Georeferenced Overlapping LiDAR Point Clouds. *Remote Sens.* **2015**, *7*, 17077–17096. [[CrossRef](#)]
9. Jin, S.; Su, Y.; Gao, S.; Wu, F.; Hu, T.; Liu, J.; Li, W.; Wang, D.; Chen, S.; Jiang, Y.; et al. Deep Learning: Individual Maize Segmentation From Terrestrial Lidar Data Using Faster R-CNN and Regional Growth Algorithms. *Front. Plant Sci.* **2018**, *9*, 866. [[CrossRef](#)]
10. Jin, S.C.; Sun, X.L.; Wu, F.F.; Su, Y.J.; Li, Y.M.; Song, S.L.; Xu, K.X.; Ma, Q.; Baret, F.; Jiang, D.; et al. Lidar sheds new light on plant phenomics for plant breeding and management: Recent advances and future prospects. *ISPRS J. Photogramm. Remote Sens.* **2021**, *171*, 202–223. [[CrossRef](#)]

11. Thapa, S.; Zhu, F.; Walia, H.; Yu, H.; Ge, Y. A Novel LiDAR-Based Instrument for High-Throughput, 3D Measurement of Morphological Traits in Maize and Sorghum. *Sensors* **2018**, *18*, 1187. [[CrossRef](#)] [[PubMed](#)]
12. Jin, S.; Su, Y.; Gao, S.; Wu, F.; Ma, Q.; Xu, K.; Ma, Q.; Hu, T.; Liu, J.; Pang, S.; et al. Separating the Structural Components of Maize for Field Phenotyping Using Terrestrial LiDAR Data and Deep Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2644–2658. [[CrossRef](#)]
13. Côté, J.-F.; Fournier, R.A.; Egli, R. An architectural model of trees to estimate forest structural attributes using terrestrial LiDAR. *Environ. Model. Softw.* **2011**, *26*, 761–777. [[CrossRef](#)]
14. Xu, H.; Gossett, N.; Chen, B.Q. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.* **2007**, *26*, 19-es. [[CrossRef](#)]
15. Su, Z.; Li, S.; Liu, H.; Liu, Y. Extracting wood point cloud of individual trees based on geometric features. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1294–1298. [[CrossRef](#)]
16. Tang, L.; Zhang, H.; Huang, H.; Chen, C. Improved Method for 3D Reconstruction of Tree Model Based on Point Cloud Data. *Trans. Chin. Soc. Agric. Mach.* **2017**, *48*, 186–194. [[CrossRef](#)]
17. Hu, C.; Liu, X.; Ji, M.; Li, Y.; Li, P. Single Poplar Leaf Segmentation Method Based on SegNet and 3D Point Cloud Clustering in Field. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 259–264. [[CrossRef](#)]
18. Xia, C.; Wang, L.; Chung, B.-K.; Lee, J.-M. In Situ 3D Segmentation of Individual Plant Leaves Using a RGB-D Camera for Agricultural Automation. *Sensors* **2015**, *15*, 20463–20479. [[CrossRef](#)]
19. Li, D.; Cao, Y.; Shi, G.; Cai, X.; Chen, Y.; Wang, S.; Yan, S. An Overlapping-Free Leaf Segmentation Method for Plant Point Clouds. *IEEE Access* **2019**, *7*, 129054–129070. [[CrossRef](#)]
20. Duan, T.; Chapman, S.C.; Holland, E.; Rebetzke, G.J.; Guo, Y.; Zheng, B. Dynamic quantification of canopy structure to characterize early plant vigour in wheat genotypes. *J. Exp. Bot.* **2016**, *67*, 4523–4534. [[CrossRef](#)]
21. Jin, S.; Su, Y.; Wu, F.; Pang, S.; Gao, S.; Hu, T.; Liu, J.; Guo, Q. Stem–Leaf Segmentation and Phenotypic Trait Extraction of Individual Maize Using Terrestrial LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1336–1346. [[CrossRef](#)]
22. Liu, G.; Zhang, W.; Guo, C. Apple Leaf Point Cloud Clustering Based on Dynamic-K-threshold and Growth Parameters Extraction. *Trans. Chin. Soc. Agric. Mach.* **2019**, *50*, 163–169+178. [[CrossRef](#)]
23. Wu, S.; Zhao, C.; Guo, X.; Wen, W.; Xiao, B.; Wang, C. Method of fruit tree canopy leaf reconstruction based on point cloud. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 212–218. [[CrossRef](#)]
24. Koma, Z.; Rutzinger, M.; Bremer, M. Automated Segmentation of Leaves From Deciduous Trees in Terrestrial Laser Scanning Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1456–1460. [[CrossRef](#)]
25. Li, D.; Li, J.; Xiang, S.; Pan, A. PSegNet: Simultaneous Semantic and Instance Segmentation for Point Clouds of Plants. *Plant Phenomics* **2022**, *2022*, 9787643. [[CrossRef](#)] [[PubMed](#)]
26. Li, Y.L.; Wen, W.L.; Miao, T.; Wu, S.; Yu, Z.T.; Wang, X.D.; Guo, X.Y.; Zhao, C.J. Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning. *Comput. Electron. Agric.* **2022**, *193*, 106702. [[CrossRef](#)]
27. Wu, S.G.; Bao, F.S.; Xu, E.Y.; Wang, Y.-X.; Chang, Y.-F.; Xiang, Q.-L. A leaf recognition algorithm for plant classification using probabilistic neural network. In Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 15–18 December 2007; pp. 11–16.
28. Bergen, G.v.d. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *J. Graph. Tools* **1997**, *2*, 1–13. [[CrossRef](#)]
29. Gottschalk, S.; Lin, M.C.; Manocha, D. OBBTree: A Hierarchical Structure for Rapid Interference Detection. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 4–9 August 1996; pp. 171–180.
30. Hosoi, F.; Omasa, K. Estimating leaf inclination angle distribution of broad-leaved trees in each part of the canopies by a high-resolution portable scanning lidar. *J. Agric. Meteorol.* **2015**, *71*, 136–141. [[CrossRef](#)]
31. Peng, C.; Li, S.; Miao, Y.; Zhang, Z.; Zhang, M.; Li, H. Stem-leaf segmentation and phenotypic trait extraction of tomatoes using threedimensional point cloud. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 187–194. [[CrossRef](#)]
32. Gopi, M.; Krishnan, S. A Fast and Efficient Projection-Based Approach for Surface Reconstruction. In Proceedings of the Brazilian Symposium on Computer Graphics & Image Processing, Fortaleza, Brazil, 10 October 2002; pp. 179–186.
33. Będkowski, J.; Pełka, M.; Majek, K.; Fitri, T.; Naruniec, J. Open source robotic 3D mapping framework with ROS—Robot Operating System, PCL—Point Cloud Library and Cloud Compare. In Proceedings of the 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, Indonesia, 10–11 August 2015; pp. 644–649.
34. Yuan, X.; Zhao, C. A Laser Point Cloud Clustering Algorithm for Robot Navigation. *Robot* **2011**, *33*, 90–96. [[CrossRef](#)]
35. Hart, P. The condensed nearest neighbor rule (Corresp.). *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516. [[CrossRef](#)]
36. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5105–5114.
37. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
38. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [[CrossRef](#)]

39. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40. [[CrossRef](#)]
40. Kolluri, R. Provably Good Moving Least Squares. *ACM Trans. Algorithms* **2008**, *4*, 18. [[CrossRef](#)]
41. Hou, W.; Li, D.; Xu, C.; Zhang, H.; Li, T. An advanced k nearest neighbor classification algorithm based on KD-tree. In Proceedings of the 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), Chongqing, China, 10–12 December 2018; pp. 902–905.
42. Li, D.W.; Shi, G.L.; Li, J.S.; Chen, Y.L.; Zhang, S.Y.; Xiang, S.Y.; Jin, S.C. PlantNet: A dual-function point cloud segmentation network for multiple plant species. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 243–263. [[CrossRef](#)]
43. Ren, M.; Zemel, R.S. End-to-end instance segmentation with recurrent attention. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 293–301.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.