

## Article

# MIX-NET: Deep Learning-Based Point Cloud Processing Method for Segmentation and Occlusion Leaf Restoration of Seedlings

Binbin Han <sup>1,2,3,4,5</sup> , Yaqin Li <sup>5</sup>, Zhilong Bie <sup>6,7</sup>, Chengli Peng <sup>8</sup>, Yuan Huang <sup>6,7</sup>  and Shengyong Xu <sup>1,2,3,4,\*</sup>

<sup>1</sup> College of Engineering, Huazhong Agricultural University, Wuhan 430070, China

<sup>2</sup> Key Laboratory of Agricultural Equipment for the Middle and Lower Reaches of the Yangtze River, Ministry of Agriculture, Wuhan 430070, China

<sup>3</sup> Shenzhen Institute of Nutrition and Health, Huazhong Agricultural University, Shenzhen 51800, China

<sup>4</sup> Shenzhen Branch, Guangdong Laboratory for Lingnan Modern Agriculture, Genome Analysis Laboratory of the Ministry of Agriculture, Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen 51800, China

<sup>5</sup> School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China

<sup>6</sup> College of Horticulture and Forestry Sciences, Huazhong Agricultural University, Wuhan 430070, China

<sup>7</sup> Key Laboratory of Horticultural Plant Biology, Ministry of Education, Wuhan 430070, China

<sup>8</sup> Electronic Information School, Wuhan University, Wuhan 430072, China

\* Correspondence: xsy@mail.hzau.edu.cn



**Citation:** Han, B.; Li, Y.; Bie, Z.; Peng, C.; Huang, Y.; Xu, S. MIX-NET: Deep Learning-Based Point Cloud Processing Method for Segmentation and Occlusion Leaf Restoration of Seedlings. *Plants* **2022**, *11*, 3342. <https://doi.org/10.3390/plants11233342>

Academic Editors: Laura-Jayne Gardiner, Ritesh Krishna and Dominik Schmidt

Received: 12 October 2022

Accepted: 24 November 2022

Published: 1 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/>).

**Abstract:** In this paper, a novel point cloud segmentation and completion framework is proposed to achieve high-quality leaf area measurement of melon seedlings. In particular, the input of our algorithm is the point cloud data collected by an Azure Kinect camera from the top view of the seedlings, and our method can enhance measurement accuracy from two aspects based on the acquired data. On the one hand, we propose a neighborhood space-constrained method to effectively filter out the hover points and outlier noise of the point cloud, which can enhance the quality of the point cloud data significantly. On the other hand, by leveraging the purely linear mixer mechanism, a new network named MIX-Net is developed to achieve segmentation and completion of the point cloud simultaneously. Different from previous methods that separate these two tasks, the proposed network can better balance these two tasks in a more definite and effective way, leading to satisfactory performance on these two tasks. The experimental results prove that our methods can outperform other competitors and provide more accurate measurement results. Specifically, for the seedling segmentation task, our method can obtain a 3.1% and 1.7% performance gain compared with PointNet++ and DGCNN, respectively. Meanwhile, the  $R^2$  of leaf area measurement improved from 0.87 to 0.93 and MSE decreased from 2.64 to 2.26 after leaf shading completion.

**Keywords:** point cloud segmentation; point cloud completion; leaf area measurement; MIX-Net; seedlings; deep learning

## 1. Introduction

Phenomics is a discipline that studies the observable morphological characteristics and their change patterns exhibited by individual plants or groups under specific conditions [1]. Plant phenomics is a key technology to further explore the intrinsic genotype–phenotype–environment association, and provides technical support for genomic functional analysis, molecular breeding, and precise management of agricultural production [2,3]. However, for plants, leaves are the most important for their external morphology and physiological functions [4]. Most of the traditional leaf measurement methods use the two-dimensional projection of the leaf on the CCD plane in a two-dimensional image to generate pixel points, so as to calculate the leaf parameters [5,6]. In practice, measurements based on 2D images cause serious measurement errors because the growth pattern and natural

deformation of the plant make it impossible for the leaf to be an absolute plane. To solve this problem and achieve more accurate measurements, it is crucial to capture the morphological configuration of leaves in three dimensions. With the rapid development of sensing technology and the improvement of computational performance, we can easily accomplish rapid data acquisition and phenotype extraction on a 3D scale. For example, LiDARR [7], low-cost RGB-D depth cameras [8], and multi-view imaging techniques [9] have been widely used in 3D plant data acquisition. However, the current techniques for processing 3D plant data, LiDAR and multi-view imaging, are very time-consuming and require much manual intervention, resulting in the accumulation of large amounts of raw data [10]. In addition, these methods limit the high throughput resolution of phenotypic indicators of interest to agronomists. However, low-cost RGB-D depth cameras are widely used in mapping, 3D reconstruction, indoor robotics, gesture recognition, and object detection and recognition due to their low cost, high measurement accuracy, and fast measurement speed [11]. Nowadays, low-cost RGB-D depth cameras are also increasingly used in plant phenotyping techniques.

In the process of measuring plant phenotypes with low-cost RGB-D depth cameras, one method is to measure plant phenotypes in a complete 3D point cloud, mainly through multi-view 3D point cloud registration [12]. This method can completely eliminate occlusions and obtain high-accuracy point clouds, but the point cloud alignment is demanding and time-consuming for image processing algorithms. Moreover, there is no good solution to the flexible registration problem under the jitter condition. Another approach is to measure plant phenotypes mainly in single-view 3D point clouds by using the mapping relationship between color and depth images [13]. However, this method is a single-view 3D point cloud, and the occlusion and overlap between leaves becomes more and more serious as the plant grows during the measurement process, which leads to more and more errors in measuring phenotypes. Especially at present, with the increasing demand for high-throughput acquisition, a low-cost, accurate measurement, and high-throughput method is urgently needed. So this paper proposes a 3D leaf shading segmentation and restoration technique using a low-cost RGB-D depth camera in a single view. This technique overcomes the problem of inter-leaf occlusion and overlap in current depth camera measurements of plant phenotypes to increase the accuracy of plant phenotype measurements over longer growth periods. However, there are still many issues to be addressed in order to achieve this goal.

First is the need to address the quality of point cloud data. Specifically, the data obtained from RGB-D sensors are coarse, and outlier and hover point noisy point clouds are severe [14]. Traditional methods use radius filtering and straight-pass filtering to filter out these noisy point clouds. For example, the researchers used Kinect V2 to capture rapeseed point clouds and remove other outliers and hover point noise based on line of sight and surface normal [15]. Although it can effectively improve the quality of the point cloud, it will remove some important points. A depth image-based filtering approach has also been proposed that can filter the noise of hover points and outliers well, but this approach cannot be directly applied to 3D point clouds [12]. Considering the shortcomings of these methods, hover points are generated because the light source is refracted when injected into the edge of the object, resulting in the receiver not receiving the signal properly. Therefore, this paper is mainly based on the fact that the normal vector of normal points in the depth camera when photographing the object has a very different offset angle with respect to the camera coordinate system line of sight compared with the hover point. As well as the sparse characteristics of outlier points, the two are combined to determine the offset angle of each normal vector relative to the camera line of sight and the spatial density of the point cloud. A new method based on domain space constraint is proposed to greatly filter out the hover points and outlier noise in 3D space.

Secondly, in the process of measuring plant phenotype, we need accurate leaf segmentation results. Especially under the condition of single view cloud processing, the phenotype detection accuracy is more sensitive to the segmentation accuracy. Due to the mapping be-

tween the color image and the depth image, depth cameras often segment in the color image and then align with the depth image to obtain the target segmentation result [16]. However, the resolution of color images of mainstream depth cameras such as Azure Kinect is much higher than that of depth images, which makes the quality of point clouds converted by this alignment effect poor. Moreover, considering the single-view object, although occluded and overlapped in the two-dimensional image, there is a certain spatial distance in the three-dimensional point cloud. So in this paper, we mainly focus on the segmentation of single-view 3D point clouds captured by depth cameras. In recent years, phenotypic measurements based on 3D models have attracted more and more research [17–20]. For example, an octree algorithm has been used to divide a single-plant point cloud into many small parts, and then each part is combined into one organ for segmentation based on the spatial topology [21]. Others first segmented 2D images using distance transform and watershed algorithms, and then performed leaf segmentation before mapping the segmented images into 3D [22]. However, these aforementioned methods require a lot of human interaction, rely on empirical parameter settings, and cannot meet the requirements of high-throughput processing in plant phenotyping studies. In contrast, deep learning-based methods can automatically extract features from large data volumes through algorithm design, providing a new perspective to address these issues [23]. For example, semantic segmentation of tomato plants in greenhouses was developed using PointNet++ [24] and further estimated leaf area indices [25]. A point cloud grid segmentation algorithm has also been improved and a hybrid segmentation model has been proposed that can adapt to the morphological differences of different individuals of cotton to achieve stem and leaf separation [5]. Recently, a point cloud segmentation network with dual-neighborhood feature extraction and dual-granularity feature fusion has been proposed to achieve semantic segmentation and leaf instance segmentation of three plants simultaneously [26]. However, the experimental materials of these methods mentioned above are mostly generated by LiDAR and multi-view 3D reconstruction techniques with good point cloud quality and less occlusion between plant leaves. However, the experimental accuracy of these methods drops dramatically when dealing with single-view 3D point clouds from RGB-D depth cameras due to the presence of large amounts of occlusion and overlap.

We consider that it is because most of the feature extraction of the above methods only divide the point cloud into local point cloud blocks to enrich the extraction of local features of the point cloud. However, for the interaction between point cloud blocks, the above methods all only perform simple interaction between adjacent blocks. Subsequently, some people also use attention mechanisms to enhance this interaction, but this approach will occupy a lot of space, and there is no simple and effective attention mechanism for 3D point clouds that can well solve the segmentation of such overlapping point clouds. Therefore, we develop a point cloud segmentation method based on the U-net shape hybrid point-mixer mechanism, referred to as MIX-Net, which consists of a continuous encoder-decoder network. In the encoder, considering previous point cloud feature extraction methods, such as PointNet++ [24] and DGCNN [27], the point cloud is first transformed into point cloud blocks using the K-nearest neighbor algorithm, and the features of each point cloud block are extracted by a convolutional neural network (CNN) or a custom convolutional module. However, converting the whole point cloud into point cloud blocks is too costly, there is also a large amount of redundancy, and the interaction between point cloud blocks is missing. We propose a simpler interactive feature fusion module that samples key points in the complete point cloud using point cloud curvature sampling, and then uses the key points for K-nearest neighbors to form point cloud blocks, which reduces the computational cost and redundancy between point cloud blocks. In addition, to increase the feature interaction between point cloud blocks, we borrow the success achieved by purely linear mlp-mixer in 2D images [28]. We design a purely linear point-mixer feature interaction network. It mainly includes the feature interaction within the interactive feature fusion point cloud blocks and the feature interaction between each point cloud block. Finally, considering that this interactive feature fusion module loses some feature information, we

also use multi-resolution feature extraction to extract the deep features of the missing point clouds. In the decoder, we employ the up-sampling module [29] to incrementally generate higher resolution point clouds, and after up-sampling, we resort to the point-mixer feature interaction network to generate complete point clouds with detailed details. Finally, we not only segmented the plant stems and leaves, but also added the mean-shift clustering algorithm at the end of the network to segment the instances among the leaves. Through experimental comparison, we found that this feature extraction and interaction approach achieves good results when dealing with plants with occlusions and overlaps.

A final challenge is the completion of missing leaves in segmented plant leaves. Although this technique is an emerging field in the application of 3D phenotypes to plants, it has been an important research problem in the graphics and vision community. For example, Poisson reconstruction has been used to complete holes in the surface of objects, but this method is characterized by a small patching area [30]. The geometric symmetry of the object is used to complete the complete object, but this method is characterized by a low quality of completion [31]. The above traditional methods perform poorly in the plant-completion task because they can only handle simple missing data and are less effective for missing plant leaves due to their varying angles and degrees of missingness. In recent years, 3D point cloud completion methods based on deep learning have achieved great success, which provides insights to solve the plant data problem. For example, a voxel-based grid algorithm has been developed to repair incomplete input data. However, the voxel-based approach is limited by its resolution, as the computational cost increases significantly with the resolution [32]. There are also global features learned first from a partial input point cloud to produce a rough complete point cloud and generate more details by collapsing the decoder operation [33]. Recently, researchers have proposed a point cloud fractal network for repairing incomplete point clouds using partial point clouds as input to keep the space of the original part unchanged and output only the missing part of the point cloud instead of the whole object [34]. Although the deep learning-based point cloud completion method has made some research progress, it still faces some challenges such as large computation and low resolution, and it is difficult to cope with the missing leaves in plants due to various missing angles and different degrees of missingness. Therefore, we would like to propose a new method that is efficient, stable, and applicable to plant leaf restoration. Firstly, for various missing angles and different degrees of missing in plants, we adopt a self-supervised learning training approach by using existing intact leaves, setting 14 missing angle viewpoints (8 vertices of squares and 6 face centers of squares) in 3D space by 3D squares, finding the distance from the viewpoints to the leaves, and removing the distance from the viewpoints to the leaves by different degrees of the missing set (15%, 25%, 50%). The nearest distance from the viewpoint to the leaf is removed to generate the missing leaf data. The network structure, we use the same network structure as the plant leaf segmentation, only in the last layer of the network structure and the loss function is different. We found that MIX-Net can also be well adapted to the plant leaf completion task and achieved good experimental results.

In conclusion, the current research methods cannot achieve satisfactory results in the tasks of denoising, plant leaf segmentation, and plant leaf completion for single-view 3D point clouds of depth cameras. To this end, this paper uses seedlings as the experimental object and first proposes a neighborhood spatial constraint method using a combination of spatial density of point clouds and differences in the angle of normal vectors relative to the camera view offset. In the filtering process of the seedling point cloud, not only can the hover points and outliers around the seedling point cloud be filtered out, but also relatively small details, such as relatively thin and narrow stems, can be retained. Subsequently, we also developed a plant point cloud segmentation and plant leaf completion method based on a U-net shape hybrid point-mixer mechanism, referred to as MIX-Net. This method consists of two main components: (1) a neighborhood aggregation strategy, which mainly transforms a complete point cloud into a sequence of point cloud blocks; (2) a point-mixer mechanism, which allows for enrichment within and between point cloud

blocks. To demonstrate the effectiveness of our method, we constructed a dataset of single-view seedling point clouds containing real labels for seedling segmentation and seedling leaf completion tasks. Experimental results show that the method not only balances the two tasks of plant segmentation and plant leaf completion well, but also both tasks obtain satisfactory performance on various common datasets. Moreover, in experiments on seedlings with occlusion and overlap, the method was able to separate stems and leaves under occlusion and to complete the missing leaves. In summary, our method provides a critical solution for inter-leaf shading and overlap in depth camera-based plant phenotyping studies. It makes it possible to achieve high-throughput acquisition and high-precision measurement of plant phenotypes using depth cameras.

## 2. Materials and Methods

### 2.1. Experimental Materials

The experimental subjects of this paper were typical melon seedlings, including watermelon seedlings (zaojia 8424), pumpkin seedlings (Jingle Fengjia), and cucumber seedlings (Jinchun No. 2). Samples were grown in the greenhouse of the Central China Branch of the Vegetable Crop Improvement Center of Huazhong Agricultural University from January 2021 to March 2021, and phenotypes were determined at the Key Laboratory of the Ministry of Horticultural Plant Biology. Seedlings were soaked in warm water, removed and drained, wrapped in gauze, and placed in a 28 °C thermostat for germination, and sown in 50-hole cavity trays. The mass ratio of grass charcoal, vermiculite, and perlite in the seedling substrate was 3:1:1, and Yara miaole compound fertilizer (1.0 kg/m<sup>3</sup>) was added to the substrate before sowing. After that, seedlings were cultivated in an artificial climate chamber at a diurnal temperature of 28 °C–18 °C and humidity of 65–85%, and seedlings were sprayed 1000 times with lairui Seedling Compound Fertilizer No. 1 after the 1-leaf-1 stages and 800 times with Yara Seedling Compound Fertilizer No. 1 until the end of 3-leaf-1 stages.

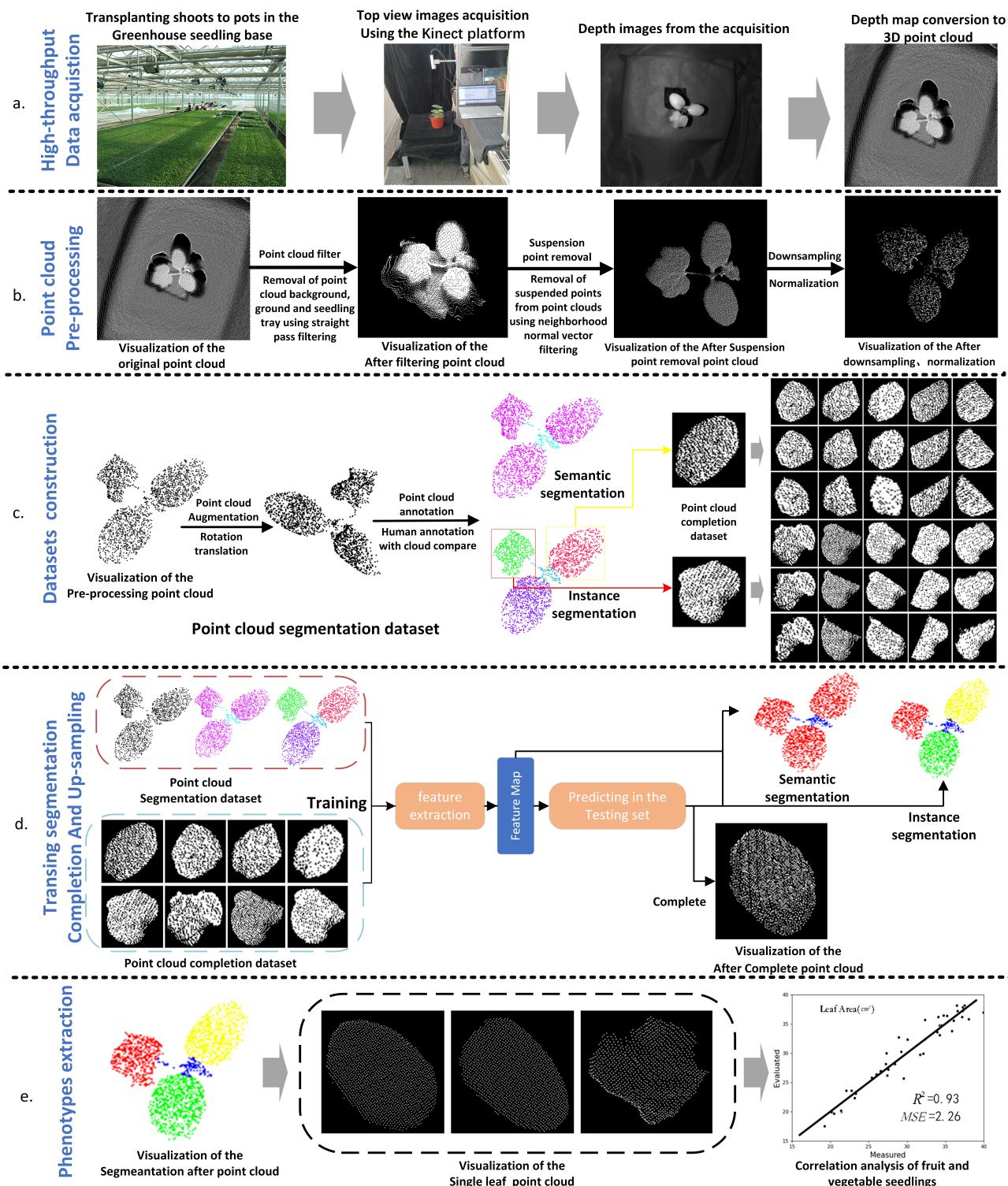
### 2.2. Mix-Net Based Seedling Point Cloud Processing Method

#### 2.2.1. Overview

Figure 1 shows the flow of the method in this paper with watermelon seedlings as an example. The method consists of five main parts: high-throughput data acquisition of seedlings, point cloud preprocessing, datasets construction, point cloud segmentation and completion, and leaf area calculation.

#### 2.2.2. Data Acquisition

Five trays (160 plants in total) of watermelon seedlings, cucumber seedlings, and pumpkin seedlings were grown for algorithm design and validation experiments using the method in Section 2.1, and destructive experiments were taken at the 1-leaf-1 stages, 2-leaf-1 stages, and for 3-leaf-1 stages phenotypic algorithm validation. High-throughput data acquisition was performed by using a semi-automatic image acquisition platform for single seedlings. It can be observed that a depth camera is mounted directly above the seedlings, and the depth camera is connected to an external computer. Target plant seedlings grown in the intelligent greenhouse are transplanted into pots and then placed in the instrument where Azure Kinect is deployed. Image acquisition and processing algorithms were developed using Azure Kinect SDK 1.4.1, Microsoft Visual Studio 2019, Window10 OS, and Tesla P100 GPU. Using this software system, 1024 × 1024 depth images can be acquired. Image acquisition was performed in a room with natural light. For the seedlings after image capture, the hand-picked flattened leaves were scanned using an Epson Expression 12000XL scanner to obtain leaf area data.



**Figure 1.** Flowchart of the procedure in this paper. (a) High-throughput seedling data acquisition using Kinect. (b) Point cloud pre-processing includes point cloud filtering, point cloud down-sampling, and normalization. (c) Annotation of data, data enhancement, and missing point cloud datasets construction using Cloud compare software. (d) Semantic segmentation of seedlings and missing leaf completion by MIX-Net. (e) Phenotype extraction using organ semantic segmentation and missing completion results.

### 2.2.3. Point Cloud Preprocessing

The point clouds collected by the Azure Kinect platform are dense, containing approximately 600,000 to 1 million points per plant. However, it contains a large amount of background noise. Hence, to ensure the accuracy and integrity of the data, the original point cloud needs to be processed by background removal and point cloud filtering before being used in subsequent steps. As shown in Figure 1b, the processing steps are as follows.

(1) Straight-pass filtering to filter out the background. For the invalid background beyond the seedlings, thresholds of 0.5 m, 0.5 m, and 0.7 m in length, width, and height are selected for direct-pass filtering since the camera height is known. Subsequently, flat ground is fitted using least squares to split the ground from the seedlings, thus removing the ground. Finally, for the seedling tray, the ground is advanced in the opposite direction of the z-axis by 0.11–0.13 as the cutting point, and the seedling tray is removed using direct-pass filtering on the z-axis.

(2) Point cloud filtering is based on the neighborhood space constraint. It includes the following steps.

- The original point cloud is filtered by (1) to obtain the point cloud containing only the plant area.
- Set a threshold  $N$ , find  $N$  neighborhoods around each centroid using KNN, and find the average value  $D$  of the Euclidean distance between the centroid and the neighborhoods.
- The angle  $W$  between the normal vector and z-axis is solved by fitting the plane with least squares to predict the normal vector of each centroid through the set neighborhood threshold  $N$ .
- Repeat the above operations First and second, if  $D \geq d$  or  $W \geq c$ , it is judged to be a hover point, and the point is deleted. Iterate through the whole point cloud to eliminate all the hover points.

Through comparison experiments, it is found that the best filtering effect is achieved when the parameter  $N$  is 12,  $d$  is 0.0034 and  $c$  is  $60^\circ$ . The setting of parameters is entirely based on the adjustment of the algorithm, independent of the parameters of the camera and the external shooting environment of the plant. Compared with the traditional point cloud filtering method, this method can not only remove the suspended points well, but also retain a large amount of point cloud details.

### 2.2.4. Datasets Construction

Point cloud data were collected from 50 samples of each of three types of melon seedlings (watermelon seedlings, cucumber seedlings, and pumpkin seedlings), covering the 1-leaf-1 stages, 2-leaf-1 stages, and 3-leaf-1 stages of the seedlings. The point clouds were then annotated using Cloud Compare software, and the annotation enhanced the data to four times the original size. A total of 600 seedling point clouds were obtained for the three types of seedlings, which constituted a point cloud segmentation data set. The complete leaf point clouds segmented by Mix-Net were generated by the missing point cloud generation method and formed 1800 point cloud pairs together with the complete point clouds as the point cloud complementary data set, as shown in Figure 1c. The division of the datasets is represented in Table 1.

(1) Data augmentation. Considering that the rotation-translation invariance and scale invariance of the point cloud, the seedling point cloud is subjected to some random panning in  $[-0.2, 0.2]$ , random anisotropic scaling in  $[0.67, 1.5]$  changes to increase the training data.

(2) Point cloud annotation. In this study, cloud compare was used to annotate the training data for stem and leaf segmentation. The stems and leaves of seedlings were first separated by entering the crop tool in the seedling point cloud selection software. For semantic segmentation, manual interactions were given to two different classes of scalar color information. Stem points were marked as 0 and leaf points were marked as 1. For the instance segmentation stem points were marked as 0 and each leaf was marked with a different marker. It takes only about 30 s to mark each seedling point cloud using

Cloud Compare software. This is very efficient due to the small size and clear structure of the seedlings.

(3) Missing point cloud generation. Firstly, for various missing angles and different degrees of missingness of plant leaves, we adopt a self-supervised learning training approach by using existing intact leaves, setting 14 missing angle viewpoints (8 square vertices and 6 square face centers) in 3D space by 3D square, finding the distance from the viewpoints to the leaves, and removing the nearest distance from the viewpoints to the leaves by different degrees of missingness set (15%, 25%, 50%) in order to generate missing leaf data. This approach generates missing point clouds similar to the missing occlusion between leaves and allows control of viewpoints as well as radii to simulate more types of missing occlusion. It has been experimentally verified that this missing approach can effectively fill in the occluded leaves.

**Table 1.** Training datasets and testing datasets settings.

	Number of Training Point Clouds	Number of Testing Point Clouds	Points	Number of Training Point Clouds after Augmentation	Number of Testing Point Clouds after Augmentation
Number of seedlings point cloud	130	20	2048	520	80
Number of leaf point clouds	500	100	2048	1800	300

## 2.2.5. MIX-NET Network for Segmenting and Completing Point Clouds

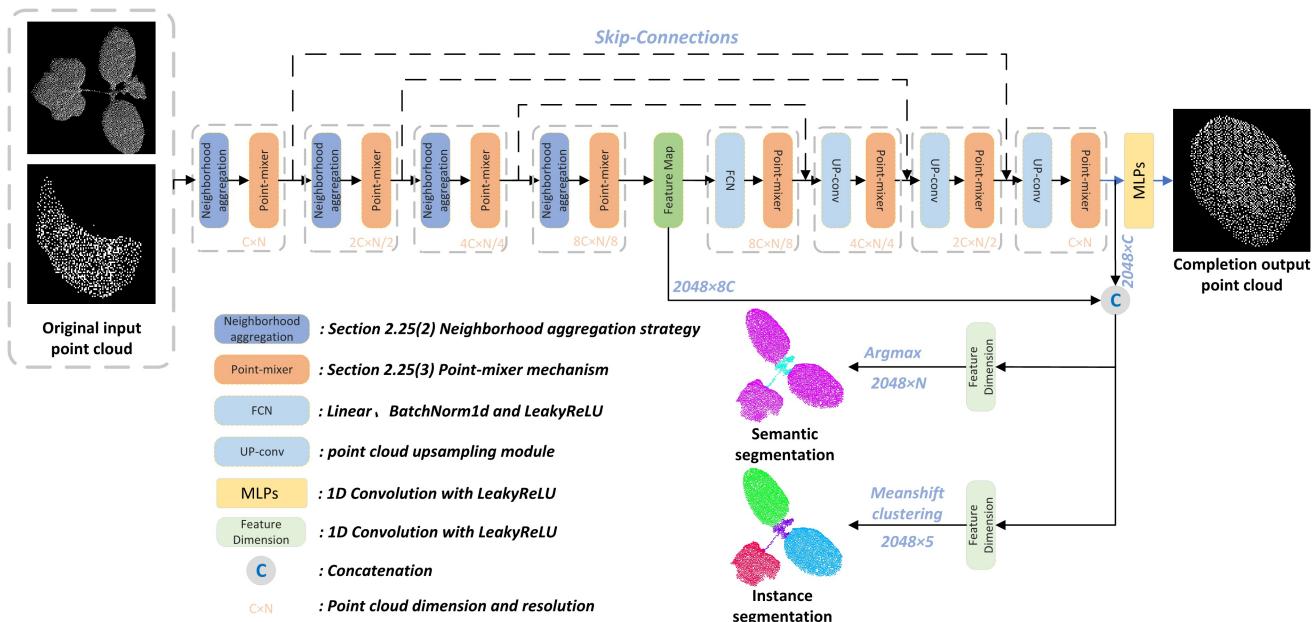
**Encoder.** The overall structure of MIX-Net's encoder is shown in the left half of Figure 2. The aim is to encode the input points into a new high-dimensional feature space. By employing a similar approach to the neighbor point aggregation mentioned in PointNet++ and PointCNN [35], the features of the point cloud are transformed into a new higher dimensional feature space, which characterizes the semantic affinity between neighbor points and serves as the basis for various point cloud processing tasks. The embedded features are then fed into the point-mixer module to learn the rich semantics within each neighbor point and the rich semantic and discriminative representations between individual neighbor points. To obtain richer point cloud features, the encoder uses multi-resolution point cloud feature extraction with 2048, 1024, 512, and 256 resolutions and neighbor points extracted in 32, 16, 8, 4, and point-mixer feature processing module dimensions of 1024, 512, 256, and 128.

**Decoder.** The decoder takes the final feature vector as input and aims to output  $M \times 3$  to represent the complete 3D point cloud shape, as shown in the right half of Figure 2. To generate higher quality complete 3D point clouds, based on FPN [36], we propose a complete progressive point cloud generation approach with the idea of generating 3D point clouds progressively from low to high resolution, such that primary, secondary, and detailed points will be predicted from layers of different feature depths. The primary and secondary points will try to match their corresponding feature points, gradually increase the number of points by interpolation up-sampling [29], and generate their high-dimensional feature maps, which will be decoded by the point-mixer module to propagate the overall geometric information to the final detailed points. In the whole process of point cloud complementation, the output point cloud resolutions of the four stages are 256, 512, 1024, and 2048; the dimensions are 1024, 512, 256, and 128.

**Point cloud classification.** We use a classification network using MIX-Net to classify a point cloud  $P$  into  $N_C$  classes of objects. The features map is fed to the classification decoder. It consists of two cascaded feed-forward neural networks LBRs (combining linear, batch norm (BN), and LeakyReLU layers), each with a Dropout probability of 0.5. A linear layer is finally used to predict the final classification. Each category scores  $\mathcal{C} \in \mathbb{R}^{N_C}$ . The category label of the point cloud is determined as the category with the maximum score.

**Point cloud segmentation.** The segmentation point cloud task is to divide it into several parts. A part label must be predicted for each point. To learn a general model applicable to various objects, we also encode the object category vector and connect it to the features map. The structure of the final output is essentially the same as that of the classification network. Then, the segmentation score  $\mathcal{S} \in \mathbb{R}^{N \times N_s}$  for each point of the final output point cloud is predicted. Finally, the label with the maximum score for each point is also identified as the label for that segment. For instance, in segmentation, the features are concatenated and then reduced to five dimensions by a feature dimension module (1D Convolution with LeakyReLU). Then the class of each instance is predicted by the mean-shift clustering algorithm.

**Point cloud completion.** We use the same network architecture as in point cloud segmentation. The difference is that in the process of generating the complete point cloud, the feature map of the same resolution in the encoder is fused with the interpolated feature maps to keep the structure of the input missing point cloud unchanged during the decoding work, after which the point-mixer is used to process the features, and then MLP is used to generate the 3D coordinates of the point cloud for each resolution.



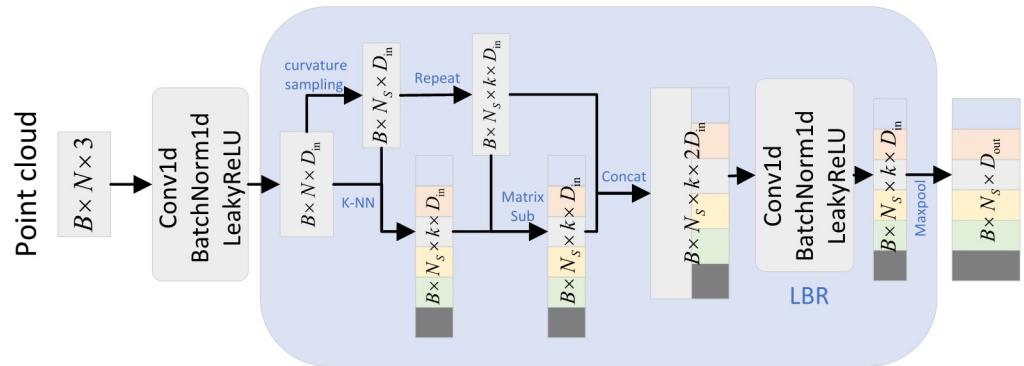
**Figure 2.** The complete structure of MIX-Net. The **left** part of the figure is the encoder, and the **right** part is the decoder. The neighborhood aggregation module is the neighborhood aggregation strategy proposed in Section 2.2.6. The point-mixer is the multi-level feature fusion module proposed in Section 2.2.7. The UP-conv module is the PointAtrousGraph, the point cloud up-sampling module proposed in the paper.

### 2.2.6. Neighborhood Aggregation Strategy

In most previous works, encoders are mostly used for feature extraction with multi-layer perception (MLP). However, they ignore the local neighborhood information, which is essential in the point cloud structure. We design a neighborhood aggregation strategy to enhance local feature extraction with neighborhood point embedding, as shown in Figure 3. More specifically, assume that the neighbor feature aggregation layer takes a point cloud  $P$  with  $N$  points and corresponding features  $F_n$  as input and outputs a sampled point cloud  $P_s$  with  $N_s$  points and their corresponding aggregated features  $F_s$ . First, we use the curvature sampling algorithm [37] to down-sample  $F_n$ , and generate features  $F_i$ . Then, with each point in feature  $F_i$  as the center, find the nearest  $k$  points in feature  $F_i$  to form a neighborhood  $F_{ik}$ . Finally, the output features  $F_s$  in the way shown in Equation (1):

$$F_s = MP(LBR(\text{concat}(F_i - F_{ik}, RP(F_i, k)))) \quad (1)$$

where  $MP$  is the maximum pooling operator and  $RP(x, k)$  is the operator that repeats the  $x$  vector  $k$  times to form a matrix. To extract more comprehensive features, multi-resolution point cloud feature extraction is used to extract and fuse their features at different resolutions.



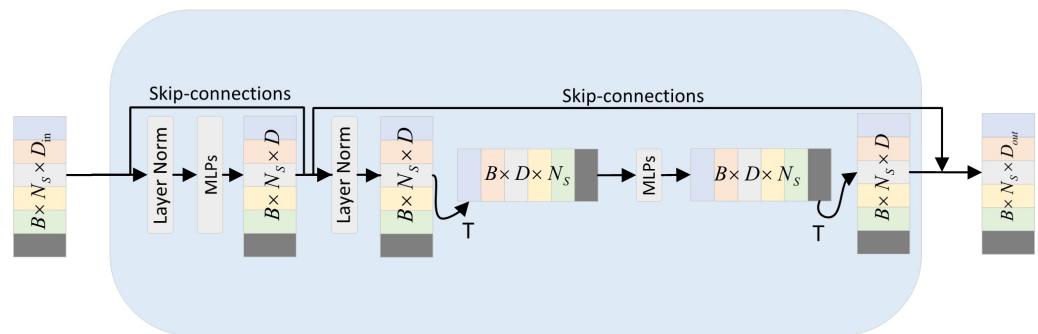
**Figure 3.** Neighborhood aggregation strategy framework diagram.

#### 2.2.7. Point-Mixer Mechanism

We propose a feature processing network that can interact within and between localities, called a point-mixer. As shown in Figure 4. The point-mixer generates  $S$  sequences of non-overlapping point cloud groups as input after passing through a local neighborhood aggregation strategy, and the point cloud group sequences are linearly projected to the desired dimension using the same projection matrix. Where point-mixer consists of multiple layers of the same size, each layer consists of two MLP blocks [28]. The first layer is the point-mixer intra-group blending MLP: it acts between the interiors of the point-mixer and maps the point-mixer sequences. The second is the inter-group hybrid MLP: it acts between point cloud groups and again maps the mapping back to the same dimensions and representation. Each MLP block contains two fully connected layers and a nonlinear operation is applied to each row of its input data tensor independently. The point-mixer process can be written in the form of Equation (2):

$$\begin{aligned} F_c &= F_s + \text{MLP}(\text{LayerNorm}(F_s)) \\ F_o &= F_c + T(\text{MLP}(T(\text{LayerNorm}(F_c)))) \end{aligned} \quad (2)$$

where  $T$  is a flip operation.  $F_c$  and  $F_o$  are tunable hidden features in the intra-group blending and inter-group blending MLP, respectively. Note that the dimension selection is independent of the number of input point cloud groups.



**Figure 4.** Point-mixer multi-level feature fusion framework.

### 2.2.8. Loss Function

**Semantic segmentation.** The softmax cross-entropy function was used as a loss function during training and is shown in Equation (3):

$$\text{Loss}_{\text{sem}} = \sum_{n=1} (-y_n \times \log(\hat{y}_n)) \quad (3)$$

where  $n$  is the total number of points in the input point cloud;  $y_n$  is the ground truth of the multi-level classification corresponding to this point cloud;  $\hat{y}_n$  is the probability of the output of each point cloud category using the softmax function. The specific formulae for are shown in Equation (4):

$$\hat{y}_n = \frac{e^{J_n}}{\sum_i e^{J_i}} \quad (4)$$

According to Equation (4) for the known  $n$ th input point, the value of  $J_n$  was calculated using Equation (5):

$$J_n = w \times q_n \quad (5)$$

where  $w$  is the weight of the network as a whole, and  $q_n$  is the input parameter for the  $n$ th point in the point cloud.

**Instance segmentation.**  $\text{Loss}_{\text{ins}}$  is given by Equation (6):

$$\begin{aligned} \text{Loss}_{\text{ins}} &= \mathbf{L}_s + \mathbf{L}_{\text{reg}} \\ L_s &= \frac{1}{I} \sum_{i=1}^I \frac{1}{N_i} \sum_{j=1}^{N_i} \left[ \max \left[ \mathbf{0}, \left\| c_i - f_j \right\|_2 - \delta_s \right] \right]^2 \\ L_{\text{reg}} &= \frac{1}{I} \sum_{i=1}^I \|c_i\|_2 \end{aligned} \quad (6)$$

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;  $c_i$  represents the center of the points belonging to the  $i$ -th instance in the current feature space; and  $f_j$  represents the feature vector of the point  $j$  in the current feature space. The parameter  $\delta_s$  defines a boundary threshold that allows the aggregation of points of the same instance.

**Point cloud completion loss.** The loss measure in the point cloud completion process represents the difference between the true complete point cloud corresponding to the missing point cloud and the predicted point cloud. Fan [38] proposed two alignment-invariant metrics to compare the difference between disordered point clouds, namely Chamfer distance (CD) and bulldozer distance (EMD). Because the bulldozer distance (EMD) occupies more memory and takes longer to calculate, while the Chamfer distance (CD) is more efficient to calculate, this paper chooses the Chamfer distance as the loss function for point cloud completion as follows Equation (7):

$$L_{\text{CD}}(S_1, S_2) = \frac{1}{2} \left( \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\| + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\| \right) \quad (7)$$

The mean nearest square distance, referred to as the Chamfer distance (CD), between the predicted point cloud  $S_1$  and the true point cloud  $S_2$  is measured using Equation (13). The progressive deconvolution completion network is a special progressive deconvolution 3D point cloud completion network in which the complete point cloud is generated in four stages with resolutions. The predicted point cloud outputs of the four stages are denoted by  $Y_1, Y_2, Y_3$ , and  $Y_4$ ; the true complete point clouds sampled from the true point cloud by IFPS to  $N/8, N/4, N/2$ , and  $N$  resolutions are denoted by  $Y_{\text{gt}}, Y'_{\text{gt}}, Y''_{\text{gt}}$ ,

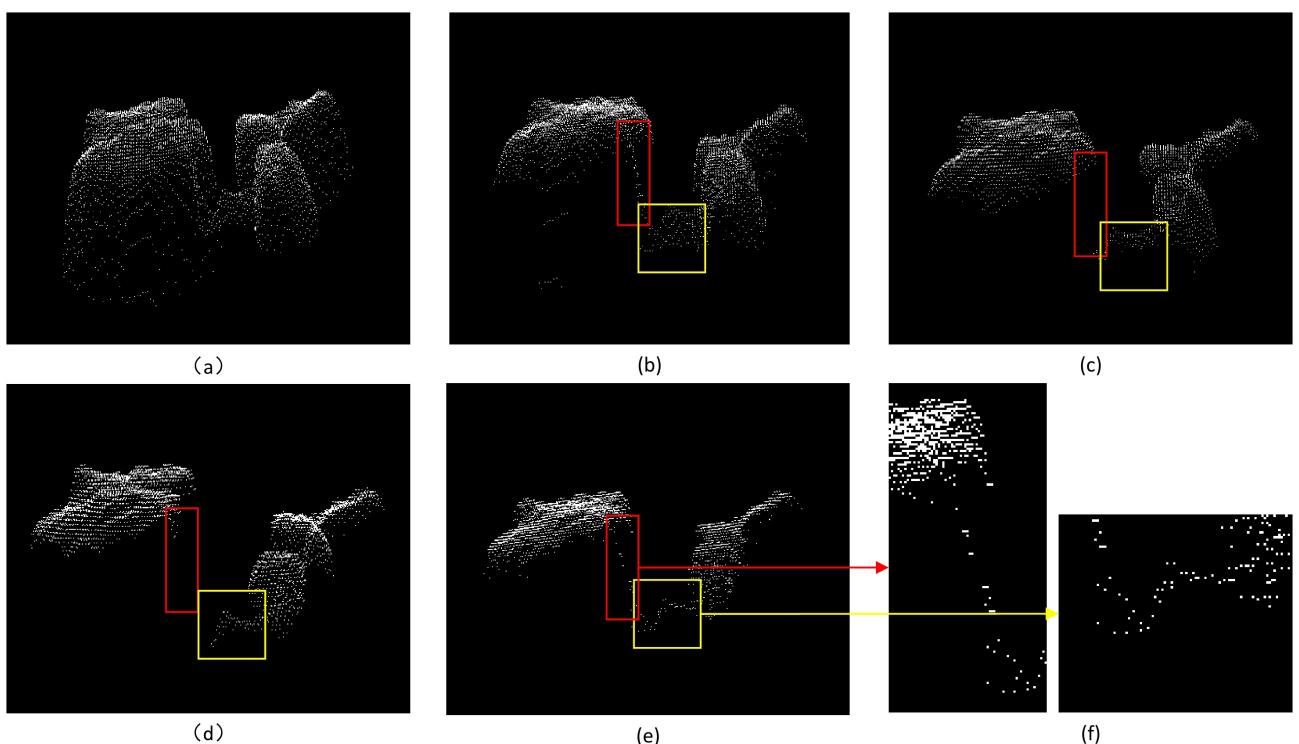
and  $Y''_{gt}$ . The Chamfer distances (CD) of the four stages are denoted by  $d_{CD_1}$ ,  $d_{CD_2}$ ,  $d_{CD_3}$ , and  $d_{CD_4}$ . The complete loss function for the training process is shown in Equation (8):

$$L_{com} = d_{CD_1}(Y_1, Y_{gt}) + d_{CD_2}(Y_2, Y'_{gt}) + d_{CD_3}(Y_3, Y''_{gt}) + d_{CD_4}(Y_4, Y'''_{gt}) \quad (8)$$

### 3. Results

#### 3.1. Point-Cloud Noise Removing

The proposed method is compared with statistical filtering, radius filtering, and domain maximum filtering for two-dimensional depth maps. The experimental results are shown in Figure 5b–e, respectively. As one can observe, all three methods result in incomplete filtering if a small radius range is set, as shown in the yellow box in Figure 5b. However, if the radius is large, the other three methods will delete some important points, as shown in the red boxes in Figure 5c,d.



**Figure 5.** (a) The original point cloud with hover point noise. (b) Statistical filtering. (c) Radius filtering. (d) The domain maximum filtering of a two-dimensional depth map. (e) Our proposed filtering results. (f) denotes the local magnification of our method after filtering.

#### 3.2. Evaluation Metrics

We compare our MIX-Net with other popular competitors on two public datasets. For a fair comparison we use the same training strategy to optimize our method and competitors and use several popular metrics in point cloud classification and segmentation to evaluate the performance, including accuracy ACC (accuracy) and part-average intersection over union ratio IoU (intersection over union). In the formula, TP (true positives) means the positive class is determined as a positive class, FP (false positives) means the negative class is determined as a positive class, FN (false negatives) means the positive class is determined as a negative class, and TN (true negatives) means the negative class is determined as a negative class.

The accuracy of the  $i$ -th class of objects in  $N$  classes is shown in Equation (9):

$$ACC_i = \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \quad (9)$$

In the  $N$  class object, the intersection ratio of the  $i$  class is shown in Equation (10):

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \quad (10)$$

The average cross-merge ratio of all classes is shown in Equation (11):

$$mIoU = \frac{1}{N} \sum_{i=1}^N IoU_i \quad (11)$$

The parameter  $C$  is the number of semantic classes for the calculation of the mean precision (mPrec) and the mean recall (mRec).

$$\begin{aligned} mPrec &= \frac{1}{C} \sum_{i=1}^C \frac{|TP(\text{sem} = i)|}{|IP(\text{sem} = i)|} \\ mRec &= \frac{1}{C} \sum_{i=1}^C \frac{|TP(\text{sem} = i)|}{|IG(\text{sem} = i)|} \end{aligned} \quad (12)$$

Because the semantic classes include the stem class and the leaf class of each plant species,  $C$  is fixed at 2. 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  $|TP(\text{sem} = i)|$  represents the total number of predicted instances in semantic class  $i$ .  $|IG(\text{sem} = i)|$  represents the number of instances of the ground truth in semantic class  $i$ .

We evaluate the reconstruction accuracy by calculating the CD between the predicted complete shape and the true shape (Equation (7)). At the same time, considering the sensitivity of CD to outliers, we also use F-score to evaluate the distance between object surfaces, which is defined as the harmonic mean between precision and recall.  $EMD$  is only defined when  $S_1$  and  $S_2$  have the same size in Equation (13):

$$\mathcal{L}_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \|x - \phi(x)\|_2 \quad (13)$$

where  $\phi$  is a bijection.

The correlation coefficient ( $R^2$ ) and mean square error ( $MSE$ ) were calculated to compare the results, which can be calculated using Equation (14):

$$\begin{aligned} R^2 &= 1 - \frac{\sum_{l=1}^m (v_l - v'_l)^2}{\sum_{l=1}^m (v_l - \bar{v}_l)^2} \\ RMSE &= \sqrt{\frac{1}{m} \sum_{l=1}^m (v_l - v'_l)^2} \end{aligned} \quad (14)$$

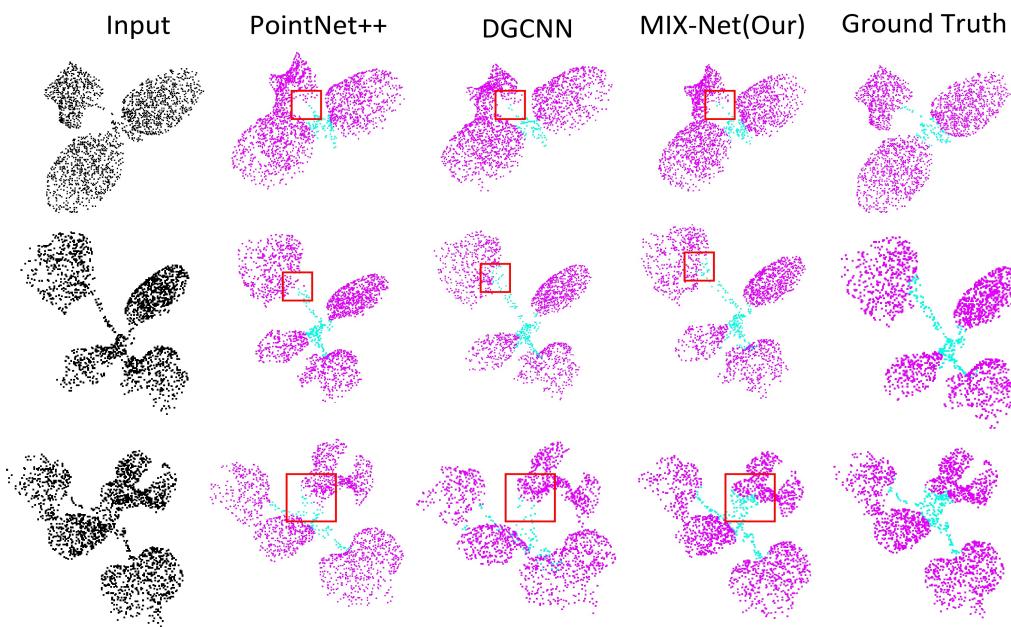
where  $m$  denotes the number of objects to be compared;  $v_l$  indicates the value of the manual measurement result;  $v'_l$  denotes the values of the phenotypic parameters extracted from the segmentation results according to the MIX-Net model;  $\bar{v}_l$  indicates the mean of manual measurement results.

### 3.3. Effectiveness of MIX-Net Network on Seedling Datasets

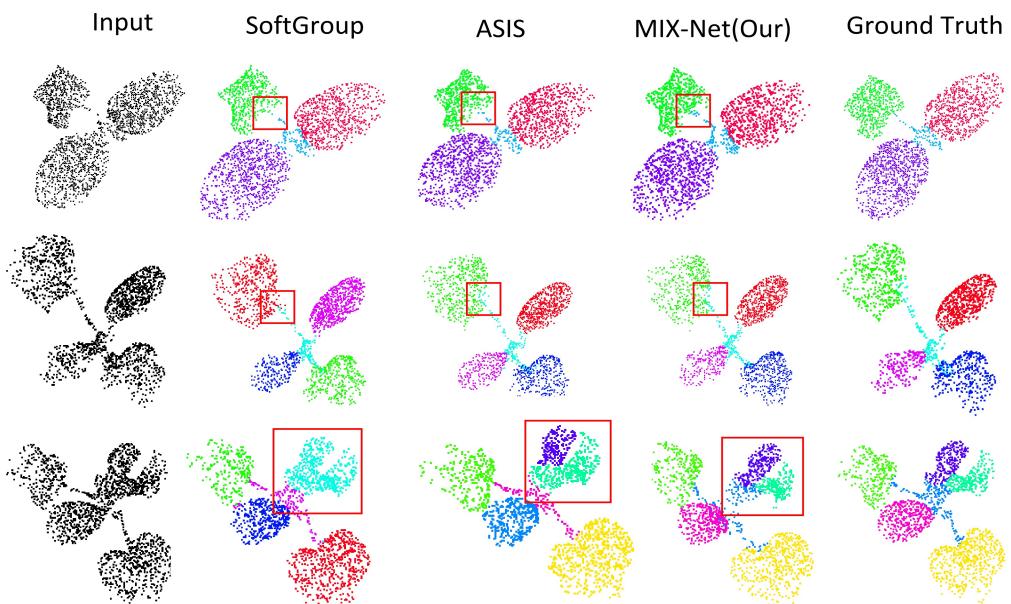
The experimental performance of MIX-Net on the seedling point cloud datasets was evaluated and compared with other methods in a comprehensive manner. For all network models, the batch size was 32, and each network was trained 250 times individually. The initial learning rate was 0.01, and a cosine function was used to decay the learning rate to adjust the learning rate for each period.

### Results of Seedling Leaf Segmentation

Experiments were conducted using MIX-Net on the seedling point cloud semantic segmentation datasets and compared with PointNet++ and DGCNN networks. The experimental results are shown in Table 2 below, and MIX-Net improved by 3.1% and 1.7%, respectively. The semantic segmentation example of seedlings segmented by MIX-Net is shown in Figure 6. Additionally, the results of the instance segmentation of seedlings compared with Soft-Group and ASIS are shown in Table 3 and Figure 7 below.



**Figure 6.** The qualitative semantic segmentation comparison on the three species. DGCNN and PointNet++ are compared with our MIX-Net. The parts with segmentation errors are highlighted by red dotted circles.



**Figure 7.** The qualitative instance segmentation comparison on the three species. SoftGroup and ASIS are compared with our MIX-Net. The parts with segmentation errors are highlighted by red dotted circles.

**Table 2.** Semantic segmentation results of MIX-Net on seedling point cloud datasets.

Methods	Input	Points	mIoU (%)
PointNet++ [24]	P	2048	91.5
DGCNN [21]	P	2048	92.9
MIX-Net (Our)	P	2048	<b>94.6</b>

**Table 3.** Instance segmentation results of MIX-Net on seedling point cloud datasets.

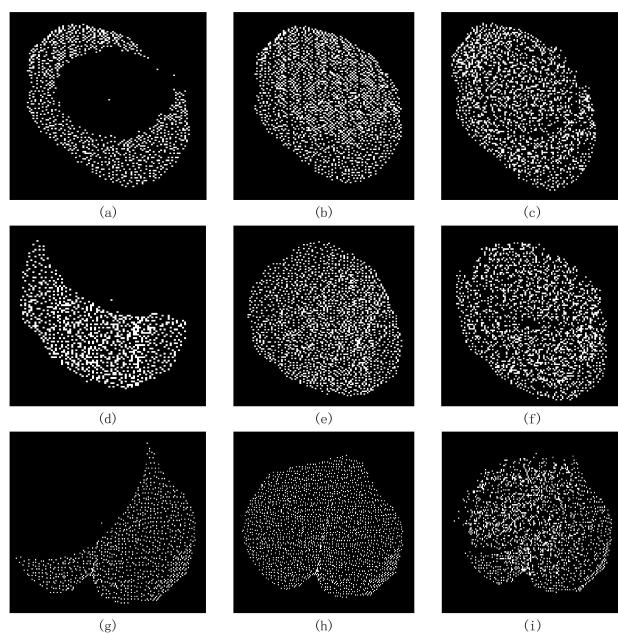
Methods	Input	Points	mPrec (%)	mRec (%)
Soft-Group [39]	P	2048	74.26	68.04
ASIS [40]	P	2048	79.13	75.64
MIX-Net (Our)	P	2048	<b>82.31</b>	<b>77.46</b>

### 3.4. Results of MIX-Net Applied to Leaf Completion under Self-Supervised Learning

We conducted experiments using MIX-Net on the seedling leaf datasets in Section 2.2.4 and compared it with its point cloud completion method. The CD and EMD evaluation metrics are given in Table 4, and the results show that MIX-Net outperforms other networks in both evaluation metrics. The results of leaf completion are given in Figure 8. It can be seen that the completion result ensures the original leaf structure remains unchanged while the output leaf is more uniform.

**Table 4.** Leaf completion results of MIX-Net on seedling point cloud leaf datasets.

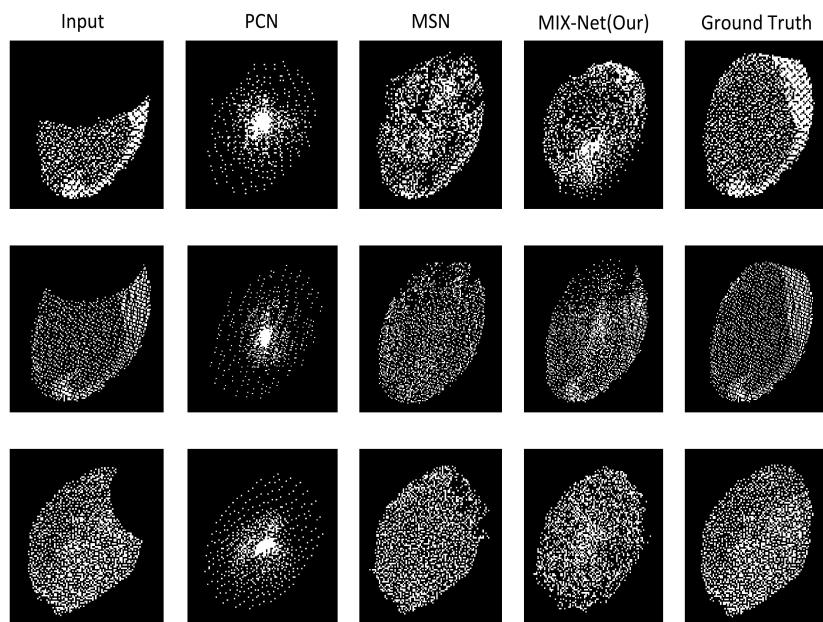
Methods	Input	Points	m-Value	CD × 10 <sup>3</sup>	EMD
PCN [33]	P	2048	50%, 25%, 15%	1.947	0.106
MSN [41]	P	2048	50%, 25%, 15%	<b>0.870</b>	0.072
PF-Net [42]	P	2048	50%, 25%, 15%	1.947	—
Vrc-Net [34]	P	2048	50%, 25%, 15%	1.783	0.107
MIX-Net (Our)	P	2048	50%, 25%, 15%	1.679	<b>0.071</b>



**Figure 8.** Representation of the complete results of the point cloud leaf datasets on MIX-Net, where (a,d,g) are the missing point cloud leaves from the input network, (b,e,h) are the real complete leaves, and (c,f,i) are the predicted outputs of the MIX-Net network.

### 3.5. Results of MIX-Net Applied to Leaf Completion under Supervised Learning

In the above experiments, the training process is self-supervised, but the actual leaf point cloud completion process is supervised learning, and since the missing leaves are extracted from reality, so to evaluate the point cloud completion capability of MIX-Net on supervised learning, the same experiment was conducted on the seedling leaf datasets in Section 2.2.4, where the missing leaf point clouds were generated, normalized, and then formed a leaf point cloud pair with the complete leaf. The results on supervised learning are given in Table 5 below, and the complete results for missing 50%, 25%, and 15% leaves are given in Figure 9 below.



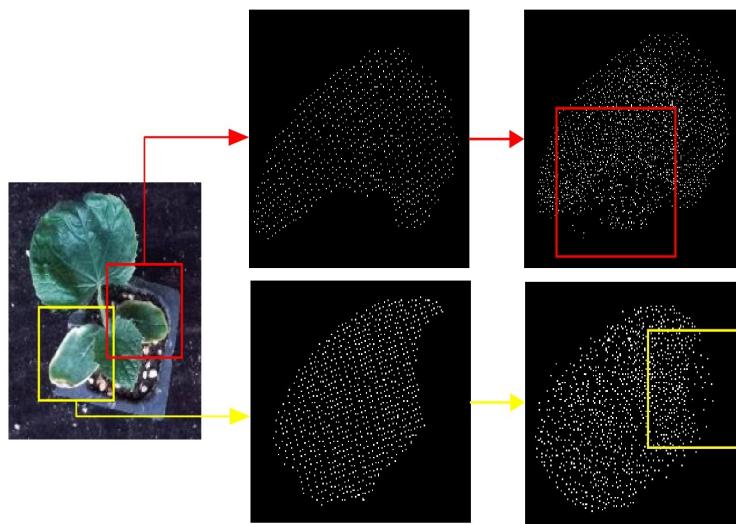
**Figure 9.** Supervised training results of MIX-Net on the point cloud leaf dataset. In the experiments, the predicted outputs of various methods after supervised training after missing 50%, 25%, and 15%.

**Table 5.** The results on supervised learning of MIX-Net on seedling point cloud leaf datasets.

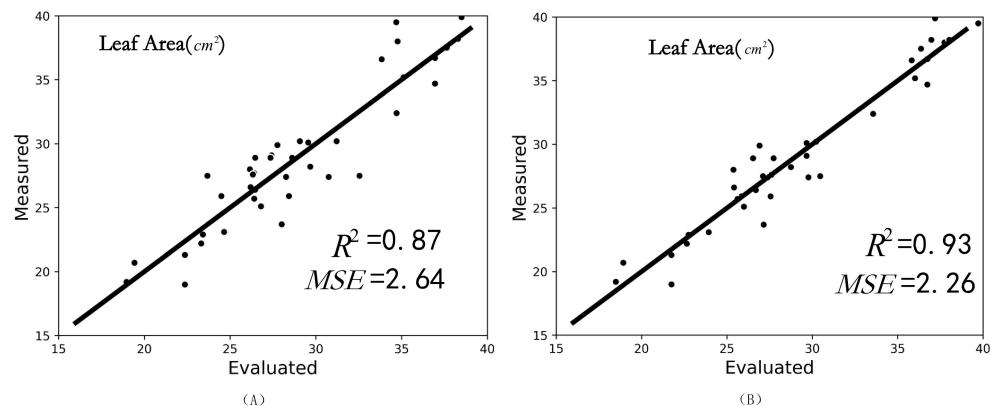
Methods	Input	Points	m-Value	CD × 10 <sup>3</sup>	EMD
PCN [33]	P	2048	50%, 25%, 15%	1.773	0.113
MSN [41]	P	2048	50%, 25%, 15%	1.914	0.065
MIX-Net (Our)	P	2048	50%, 25%, 15%	<b>1.276</b>	<b>0.063</b>

### 3.6. Nondestructive Leaf Area Measurement Results Using MIX-Net

To verify that our experimental results are helpful in realistic phenotypic measurements, we selected 40 seedlings with occlusion, as shown in Figure 10 below, and then used MIX-Net to isolate the missing leaves of the seedlings, and to complete them. Figure 11 gives the correlation coefficient between the true leaf area (by leaf area meter) and the leaf area before repair with  $R^2 = 0.87$ ,  $MSE = 2.64$ . The correlation coefficient between the true leaf area (by leaf area meter) and the leaf area after repair after completing is  $R^2 = 0.93$ ,  $MSE = 2.26$ . The presence of occlusion is particularly serious, we select the leaves which are severely occluded, and from the result of the patching, we can get complete and uniform leaves, and our method provides help for the nondestructive testing of the whole tray of seedlings.



**Figure 10.** Complementary results of MIX-Net in real phenotypic measurements, where indicates the presence of occlusion in phenotypic measurements of seedlings and our corresponding complementary results.



**Figure 11.** Correlation analysis after leaf area measurement, (A) indicates the correlation analysis between the leaf before restoration and the real leaf; (B) indicates the correlation analysis between the restored and the real leaf.

#### 4. Discussion

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

##### 4.1. Point Cloud Classification Results on the Modelnet40 Datasets

The ModelNet40 datasets [43] contain 12,311 CAD models in 40 object classes; it is widely used for point cloud shape classification benchmarks. For a fair comparison, we use 9843 objects from the office for training and 2468 objects for evaluation. The same sampling strategy as PointNet [44] was used, sampling each object uniformly to 1024 points. During training, no data augmentation or voting methods were used during testing. For all network models, the batch size was 32, and each network was trained 250 times individually. The initial learning rate was 0.01, and a cosine function was used to decay the learning rate to adjust the learning rate for each period. The experimental results are listed in Table 6. Compared with PointNet and PCT, MIX-Net improved by 4.2% and 0.2%. The overall accuracy of MIX-Net was 93.4%. It is worth mentioning that our network currently does not consider normal vectors as input.

**Table 6.** Comparison with state-of-the-art methods on the ModelNet40 classification datasets. Precision implies overall accuracy. All cited results are taken from the cited papers. P denotes the number of points, and N denotes the normal.

Methods	Input	Points	Accuracy (%)
PointNet++ [24]	P	1024	90.7
PointNet++ [24]	P, N	1024	91.9
PointCNN [35]	P	1024	92.5
DGCNN [27]	P	1024	92.9
PCT [45]	P	1024	93.2
MIX-Net (Our)	P	1024	<b>93.4</b>

#### 4.2. Point Cloud Segmentation Results on the ShapeNet-Part Dataset

We experimentally evaluate the ShapNet-Part dataset [46], which contains 16,880 3D models trained to test segmentation from 14,006 to 2874. It has 16 object classes and 50 part labels, and each instance contains no less than two parts. All models were down-sampled to 2048 points, preserving the individual point annotations. The models have a batch size of 16, a training count of 250, and a learning rate of 0.001. Table 7 shows the segmentation results for each type of network. The evaluation metric used is part-average intersection over union. The results show that our MIX-Net improves by 2.0% over PointNet. MIX-Net reaches 85.7%.

**Table 7.** Comparison using ShapeNet-Part segmentation dataset. mIoU denotes the average intersection over union. All cited results are taken from the cited papers.

Methods	Input	Points	mIoU (%)
PointNet++ [24]	P	2048	85.1
DGCNN [27]	P	2048	85.2
MIX-Net (Our)	P	2048	<b>85.7</b>

#### 4.3. Point Cloud Completion Validated on a ShapeNet-Part Dataset

To train our model, we used 13 different objects in the ShapeNet-Part of the benchmark dataset. The total number of shapes is 14,473 (11,705 for training and 2768 for testing). All input point cloud data are centered at the origin, and their coordinates are normalized to  $[-1, 1]$ . Ground truth point cloud data were created by sampling 2048 points uniformly on each shape. Incomplete point cloud data were generated by the missing point cloud generation method. We control the parameters to get different numbers of missing points. When comparing our method with other methods, incomplete point clouds with 50% of the original data missing are set up for training and testing. For all network models, the batch size is 16, and each network is trained 100 times separately. The initial learning rate was 0.0001, and a cosine function for learning rate decay was used to adjust the learning rate for each period. Table 8 shows the completion results for each type of network. The evaluation metrics used are CD distance and F-score@1%. The results show that our MIX-Net achieves optimal results in both CD and F-score@1%.

**Table 8.** Point cloud completion results (CD and F-Score@1%) on the ShapeNet-Part dataset (2048 points).

Methods	Input	Points	m-Value	CD $\times 10^3$	F-Score@1%
PCN [33]	P	2048	50%	2.929	0.29
TopNet [47]	P	2048	50%	3.805	0.38
MSN [41]	P	2048	50%	2.376	0.41
PF-Net [42]	P	2048	50%	3.037	—
Vrc-Net [34]	P	2048	50%	2.881	0.42
MIX-Net (Our)	P	2048	50%	<b>2.111</b>	<b>0.45</b>

#### 4.4. Ablation Experiments

In Table 9 we will discuss the effectiveness of the proposed framework in this paper. We will evaluate it separately in point cloud classification, point cloud segmentation, and point cloud completion. Firstly, for point cloud classification, the feature extraction ability of these modules for point clouds can be verified. Among them, we choose the encoder of PointNet++ as a baseline. Then it is combined with Nas (neighborhood aggregation strategy) and point-mixer to verify the effectiveness of the proposed framework on the modelnet40 datasets. Then for semantic segmentation and instance segmentation of point clouds, we select PointNet++ and ASIS as baselines, respectively, and then experiment on seedling point cloud segmentation datasets. Finally, we use PCN as the benchmark and then validate the effectiveness on leaf restoration on seedling leaf completion datasets. Through the experiments, we demonstrate that our proposed module achieves excellent results in various fields.

**Table 9.** The ablation analysis of MIX-Net. The checkmark stands for the use of a module. The best quantitative values are shown in bold.

	PointNet++ (encoder) [24]		Nas	Point-mixer	Accuracy (%)	
classification (modelnet40 dataset)	<input checked="" type="checkbox"/>				90.7	
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		89.4	
	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	92.7	
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<b>93.4</b>	
	PointNet++ (encoder) [24]	PointNet++ (decoder) [24]	Nas	Point-mixer	mIoU (%)	
semantic segmentation (seedling semantic segmentation dataset)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			91.5	
	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	92.4	
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	93.7	
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		91.8	
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<b>94.6</b>	
	ASIS (encoder) [40]	ASIS (decoder) [40]	Nas	Point-mixer	mPrec (%)	mRec (%)
instance segmentation (seedling instance segmentation dataset)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			79.13	75.64
	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	77.41	72.36
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	81.32	<b>79.56</b>
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		78.44	76.54
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<b>82.31</b>	77.46
	PCN (encoder) [33]	PCN (decoder) [33]	Nas	Point-mixer	CD × 10 <sup>3</sup>	EMD
Leaf completion (seedling leaf completion dataset)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			1.773	0.113
	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	1.345	0.094
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<b>1.254</b>	0.061
	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		1.493	0.108
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	1.276	<b>0.059</b>

#### 5. Conclusions

In this study, based on high-throughput data acquisition and deep neural networks, automatic segmentation and completion method for seedling 3D point clouds is proposed. The proposed method can achieve high-quality segmentation and completion from two aspects. Firstly, during the data processing we developed a new method for eliminating hover points and noise points, which can retain more detailed features while removing noise compared with traditional statistical filtering and radius filtering. Secondly, a new network named MIX-Net is proposed to achieve point cloud segmentation and completion simultaneously, which can better balance these two tasks in a more definite and effective way and ensure high performance on these two tasks. Experimental results prove that, compared with state-of-the-art methods, the average performance gain brought by our

methods on classification, seedling segmentation, and seedling leaf completion tasks are more than PCT, DGCNN, and Vrc-Net, respectively, leading to more accurate measurement performance on the leaf area phenotypes of seedlings. Furthermore, we also explored the effect of restoration in dealing with the presence of extensive occlusion in the whole tray of seedlings, which provides feasible help for future nondestructive testing of whole-tray seedlings.

**Author Contributions:** B.H.: Conceptualization, Investigation, Methodology, Visualization, Writing—original draft. Y.L.: Data curation, Software. Z.B.: Resources. C.P.: Validation. Y.H.: Funding acquisition, Supervision. S.X.: Funding acquisition, Methodology, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the National Key Research and Development Program of China (grant number 2019YFD1001900); the Fundamental Research Funds for the Central Universities (grant number BC2021201); the HZAU-AGIS Cooperation Fund (grant number SZYJY2022006); the Hubei provincial key research and development program (grant number 2021BBA239).

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

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

## References

1. Feng, L.; Raza, M.A.; Li, Z.; Chen, Y.; Khalid, M.H.B.; Du, J.; Liu, W.; Wu, X.; Song, C.; Yu, L.; et al. The influence of light intensity and leaf movement on photosynthesis characteristics and carbon balance of soybean. *Front. Plant Sci.* **2019**, *9*, 1952. [[CrossRef](#)]
2. Ninomiya, S.; Baret, F.; Cheng, Z.M.M. Plant phenomics: Emerging transdisciplinary science. *Plant Phenomics* **2019**, *2019*, 2765120. [[CrossRef](#)]
3. Liu, H.J.; Yan, J. Crop genome-wide association study: A harvest of biological relevance. *Plant J.* **2019**, *97*, 8–18. [[CrossRef](#)] [[PubMed](#)]
4. Gara, T.W.; Skidmore, A.K.; Darvishzadeh, R.; Wang, T. Leaf to canopy upscaling approach affects the estimation of canopy traits. *GIScience Remote Sens.* **2019**, *56*, 554–575. [[CrossRef](#)]
5. Fu, L.; Tola, E.; Al-Mallahi, A.; Li, R.; Cui, Y. A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosyst. Eng.* **2019**, *183*, 184–195. [[CrossRef](#)]
6. Sapoukhina, N.; Samiei, S.; Rasti, P.; Rousseau, D. Data augmentation from RGB to chlorophyll fluorescence imaging application to leaf segmentation of *Arabidopsis thaliana* from top view images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
7. Panjvani, K.; Dinh, A.V.; Wahid, K.A. LiDARPheno—A low-cost lidar-based 3D scanning system for leaf morphological trait extraction. *Front. Plant Sci.* **2019**, *10*, 147. [[CrossRef](#)]
8. Hu, C.; Li, P.; Pan, Z. Phenotyping of poplar seedling leaves based on a 3D visualization method. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 145–151. [[CrossRef](#)]
9. Wu, S.; Wen, W.; Wang, Y.; Fan, J.; Wang, C.; Gou, W.; Guo, X. MVS-Pheno: A portable and low-cost phenotyping platform for maize shoots using multiview stereo 3D reconstruction. *Plant Phenomics* **2020**, *2020*, 1848437. [[CrossRef](#)]
10. Wang, Y.; Wen, W.; Wu, S.; Wang, C.; Yu, Z.; Guo, X.; Zhao, C. Maize plant phenotyping: Comparing 3D laser scanning, multi-view stereo reconstruction, and 3D digitizing estimates. *Remote Sens.* **2018**, *11*, 63. [[CrossRef](#)]
11. Xu, H.; Hou, J.; Yu, L.; Fei, S. 3D Reconstruction system for collaborative scanning based on multiple RGB-D cameras. *Pattern Recognit. Lett.* **2019**, *128*, 505–512. [[CrossRef](#)]
12. Teng, X.; Zhou, G.; Wu, Y.; Huang, C.; Dong, W.; Xu, S. Three-dimensional reconstruction method of rapeseed plants in the whole growth period using RGB-D camera. *Sensors* **2021**, *21*, 4628. [[CrossRef](#)] [[PubMed](#)]
13. Lee, J.E.; Park, R.H. Segmentation with saliency map using colour and depth images. *IET Image Process.* **2015**, *9*, 62–70. [[CrossRef](#)]
14. Hu, Y.; Wu, Q.; Wang, L.; Jiang, H. Multiview point clouds denoising based on interference elimination. *J. Electron. Imaging* **2018**, *27*, 023009. [[CrossRef](#)]
15. Ma, Z.; Sun, D.; Xu, H.; Zhu, Y.; He, Y.; Cen, H. Optimization of 3D Point Clouds of Oilseed Rape Plants Based on Time-of-Flight Cameras. *Sensors* **2021**, *21*, 664. [[CrossRef](#)] [[PubMed](#)]
16. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 213–228.
17. van Dijk, A.D.J.; Kootstra, G.; Kruijer, W.; de Ridder, D. Machine learning in plant science and plant breeding. *Iscience* **2021**, *24*, 101890. [[CrossRef](#)] [[PubMed](#)]
18. Hesami, M.; Jones, A.M.P. Application of artificial intelligence models and optimization algorithms in plant cell and tissue culture. *Appl. Microbiol. Biotechnol.* **2020**, *104*, 9449–9485. [[CrossRef](#)]

19. Singh, A.; Ganapathysubramanian, B.; Singh, A.K.; Sarkar, S. Machine learning for high-throughput stress phenotyping in plants. *Trends Plant Sci.* **2016**, *21*, 110–124. [[CrossRef](#)]
20. Grinblat, G.L.; Uzal, L.C.; Larese, M.G.; Granitto, P.M. Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* **2016**, *127*, 418–424. [[CrossRef](#)]
21. Duan, T.; Chapman, S.; Holland, E.; Rebetzke, G.; 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](#)]
22. Itakura, K.; Hosoi, F. Automatic leaf segmentation for estimating leaf area and leaf inclination angle in 3D plant images. *Sensors* **2018**, *18*, 3576. [[CrossRef](#)] [[PubMed](#)]
23. Jiang, Y.; Li, C.; Takeda, F.; Kramer, E.A.; Ashrafi, H.; Hunter, J. 3D point cloud data to quantitatively characterize size and shape of shrub crops. *Hortic. Res.* **2019**, *6*, 43. [[CrossRef](#)] [[PubMed](#)]
24. 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 1st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
25. Masuda, T. Leaf area estimation by semantic segmentation of point cloud of tomato plants. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1381–1389.
26. 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](#)]
27. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *AcM Trans. Graph. (ToG)* **2019**, *38*, 1–12. [[CrossRef](#)]
28. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24261–24272.
29. Pan, L.; Chew, C.M.; Lee, G.H. PointAtrousGraph: Deep hierarchical encoder-decoder with point atrous convolution for unorganized 3D points. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1113–1120.
30. Kazhdan, M.; Hoppe, H. Screened poisson surface reconstruction. *ACM Trans. Graph. (ToG)* **2013**, *32*, 1–13. [[CrossRef](#)]
31. Mitra, N.J.; Pauly, M.; Wand, M.; Ceylan, D. Symmetry in 3d geometry: Extraction and applications. *Comput. Graphics Forum* **2013**, *32*, 1–23. [[CrossRef](#)]
32. Yang, B.; Wen, H.; Wang, S.; Clark, R.; Markham, A.; Trigoni, N. 3d object reconstruction from a single depth view with adversarial learning. In Proceedings of the IEEE international Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 679–688.
33. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. Pcn: Point completion network. In Proceedings of the IEEE 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 728–737.
34. Pan, L.; Chen, X.; Cai, Z.; Zhang, J.; Zhao, H.; Yi, S.; Liu, Z. Variational relational point completion network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8524–8533.
35. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, QC, Canada, 3–8 December 2018.
36. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
37. Pagani, L.; Scott, P.J. Curvature based sampling of curves and surfaces. *Comput. Aided Geom. Des.* **2018**, *59*, 32–48. [[CrossRef](#)]
38. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
39. Vu, T.; Kim, K.; Luu, T.M.; Nguyen, T.; Yoo, C.D. SoftGroup for 3D Instance Segmentation on Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2708–2717.
40. Wang, X.; Liu, S.; Shen, X.; Shen, C.; Jia, J. Associatively segmenting instances and semantics in point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4096–4105.
41. Liu, M.; Sheng, L.; Yang, S.; Shao, J.; Hu, S.M. Morphing and sampling network for dense point cloud completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11596–11603.
42. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. Pf-net: Point fractal network for 3d point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
43. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
44. Li, R.; Li, X.; Heng, P.A.; Fu, C.W. Point cloud upsampling via disentangled refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 344–353.
45. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [[CrossRef](#)]
46. Yi, L.; Kim, V.G.; Ceylan, D.; Shen, I.C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; Guibas, L. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph. (ToG)* **2016**, *35*, 1–12. [[CrossRef](#)]
47. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. Topnet: Structural point cloud decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 383–392.