agronomy

MDPI

Article

Win-Former: Window-Based Transformer for Maize Plant Point Cloud Semantic Segmentation

Yu Sun [1,2], Xindong Guo [1,3] and Hua Yang [1,*]

[1] College of Information Science and Engineering, Shanxi Agricultural University, Jinzhong 030801, China; sunyu@sxau.edu.cn (Y.S.); gxd@sxau.edu.cn (X.G.)
[2] School of Information Science and Technology, Northwest University, Xi'an 710127, China
[3] College of Computer Science and Technology, North University of China, Taiyuan 030051, China
* Correspondence: yanghua@sxau.edu.cn

Abstract: Semantic segmentation of plant point clouds is essential for high-throughput phenotyping systems, while existing methods still struggle to balance efficiency and performance. Recently, the Transformer architecture has revolutionized the area of computer vision, and has potential for processing 3D point clouds. Applying the Transformer for semantic segmentation of 3D plant point clouds remains a challenge. To this end, we propose a novel window-based Transformer (Win-Former) network for maize 3D organic segmentation. First, we pre-processed the Pheno4D maize point cloud dataset for training. The maize points were then projected onto a sphere surface, and a window partition mechanism was proposed to construct windows into which points were distributed evenly. After that, we employed local self-attention within windows for computing the relationship of points. To strengthen the windows' connection, we introduced a Cross-Window self-attention (C-SA) module to gather the cross-window features by moving entire windows along the sphere. The results demonstrate that Win-Former outperforms the famous networks and obtains 83.45% mIoU with the lowest latency of 31 s on maize organ segmentation. We perform extensive experiments on ShapeNet to evaluate stability and robustness, and our proposed model achieves competitive results on part segmentation tasks. Thus, our Win-Former model effectively and efficiently segments the maize point cloud and provides technical support for automated plant phenotyping analysis.

Keywords: 3D point cloud; maize; transformer; self-attention; semantic segmentation

Citation: Sun, Y.; Guo, X.; Yang, H. Win-Former: Window-Based Transformer for Maize Plant Point Cloud Semantic Segmentation. *Agronomy* **2023**, *13*, 2723. https://doi.org/10.3390/agronomy13112723

Academic Editor: Juncheng Ma

Received: 19 September 2023
Revised: 26 October 2023
Accepted: 27 October 2023
Published: 29 October 2023

1. Introduction

Maize (*Zea mays* L.) is one of the world's most important cereal crops [1,2]. Modern crop breeding has become essential for cultivating high-yielding and high-quality crop varieties [3]. Plant phenotyping [4,5] refers to the unique physiological and biochemical morphological parameters or traits of a plant in response to its genotype and growing environment. In modern agriculture, plant phenotyping is essential in analyzing the growth and development of individual plants. While genomic breeding research has advanced significantly, plant phenomics has hindered the high-throughput measurements of plant gene sequences in breeding works [6]. Current plant phenotyping analyses suffer from intense manual labor in measuring and evaluating plant growth. These approaches are always time-consuming and prone to human bias. Some trait derivation methods are even destructive to the plant. Developing automatic and accurate high-throughput plant phenotyping systems is urgently needed [7]. Semantic segmentation of a plant's organs is essential for high-throughput phenotyping systems, distinguishing the organ (e.g., stem, leaf, flower) from the data and assigning a category label to each point or face.

With the advance in sensing technology [8], light detection and ranging (LiDAR) [9,10], and computational performance [11], point clouds have been adopted in computer vision tasks, including robots [12] construction industries [13], and augmented reality [14]. The

3D point cloud data with accurate parameters (i.e., coordinates and depth) and superior robustness make it suitable for assessing plant morphology, growth, and biomass [15].

Most typical conventional networks craft point cloud features for semantic segmentation tasks. Jin et al. [16] presented a median normalized vector growth (MNVG) approach to segment the plant point cloud for the analysis of phenotypic traits. Elnashef et al. [17] utilized the density-based spatial clustering of applications with noise (DBSCAN) technique for maize leaf and stem segmentation. However, for high-throughput plant phenotyping analysis, the above approaches are not suitable for complex environments and require a lot of manual manipulation, which is inefficient and ineffective.

Recently, researchers have been investigating deep learning methods on 3D point clouds [18]. According to the data representation of the neural network, 3D deep learning algorithms can be classified into several taxonomies: projection-based, voxel-based, and point-based networks. Projection-based networks [19] typically project an irregular and unstructured point cloud onto regular representations like 2D images to which a standard 2D convolutional network is applied. Voxel-based networks [20] typically visualize a point cloud and apply 3D CNNs. Point-based networks directly process the raw point cloud without any data transformation and do not suffer from explicit information loss. PointNet [21] is a pioneer deep neural network with respect to the permutation invariance of points. As PointNet lacks local connections between points, Qi et al. [22] proposed PointNet++, which obtains hierarchical local features with increasing contextual scales.

The emergence of high-resolution 3D agricultural benchmark datasets and 3D deep learning methods boost the application of point clouds in botanical and agricultural research. Turgut et al. [23] used deep learning methods like PointNet and DGCNN to segment the rosebush models. The mIoU of the segmentation results of the PointNet, PointNet++, and DGCNN are 32.97%, 81.53%, and 38.29, respectively. Li et al. [24] presented a DeepSeg3DMaize network based on the PointNet model to segment the stem-leaf of maize plants. Han et al. [25] proposed a MIX-Net network for maize leaf segmentation and completion using the purely linear mixer mechanism. The results showed that the mIoU of this model surpasses PointNet++ and DGCNN. Guo et al. [26] introduced the FF-Net network for plant segmentation and classification, comprising a voxel branch and a point branch with an attention-based module. The FF-Net obtained mIoU of 80.95 on maize point cloud segmentation. The above methods have displayed high accuracy. However, the project-based methods and Voxel-based methods may lose geometric detail during projection and voxelization and incur high computing and memory costs. Because of employing the neighborhood search strategy, point-based methods' computational complexity and running time would significantly increase when processing the large-scale points.

In recent years, Transformer has been widely introduced to computer vision tasks, such as image segmentation [27,28] and tracking. Transformer's remarkable global feature learning capability, parallel operation, and order-independent approach make it ideal for 3D point cloud processing and analysis [29]. Several Transformers' backbones and architectures are presented for the classification and segmentation of point clouds [30,31]. The Transformer-based model Point Cloud Transformer (PCT) [32] designs an improved self-attention module in point processing. Dosovitskiy et al. [33] propose the Vision Transformer algorithm that partitions the image into patches. Cui et al. [34] propose a feature fusion network based on the Transformer architecture with self-attention mechanism, cross-attention, and feature fusion network. However, the application of Transformer-based algorithms for automatically segmenting 3D plant point clouds are scarce. Moreover, the farthest point searching (FPS) and the k-nearest neighbor (KNN) used by most existing networks consume a lot of time in constructing local regions.

To address the above-mentioned problems, we proposed a Window-based Transformer (Win-Former) network to semantically segment the maize plant point cloud. First, we augmented the 3D maize point cloud dataset. We then projected the maize point cloud onto a sphere surface, which allowed us to treat the points as images by reducing their dimensionality. Additionally, we introduced a novel Window Transformer to partition

points on the sphere into windows and perform local self-attention (L-SA) hierarchically in parallel. The Window-based mechanism is an efficient neighbor-search strategy that considers the windows as local neighborhoods. Finally, we introduced Cross-Window operation to link nearby windows. Extensive experiments display that the Win-Former model effectively and efficiently segments the maize point cloud and provides technical support for automated plant phenotyping analysis.

2. Materials and Methods

2.1. Data Acquisition

We utilized maize point clouds in the Pheno4D [35] dataset for the segmentation task; the data can be downloaded from www.ipb.uni-bonn.de/data/pheno4d/ (accessed on 12 December 2022). In the Pheno4D dataset, the maize plants were grown in pots in a greenhouse. Seven maize point clouds were obtained at an early stage of growth, and the plants were numbered from one to seven. Measurements were taken periodically after the first seedling sprouts appeared and lasted about two weeks. The scanning system could thoroughly scan the plant surface at various positions and angles. Eventually, 84 maize point clouds were available, while only 49 of them were annotated. The 3D maize plant points were divided into "soil", "stem", and "leaf" categories, and each leaf was labeled distinguishable from others of the same plant, as shown in Figure 1.

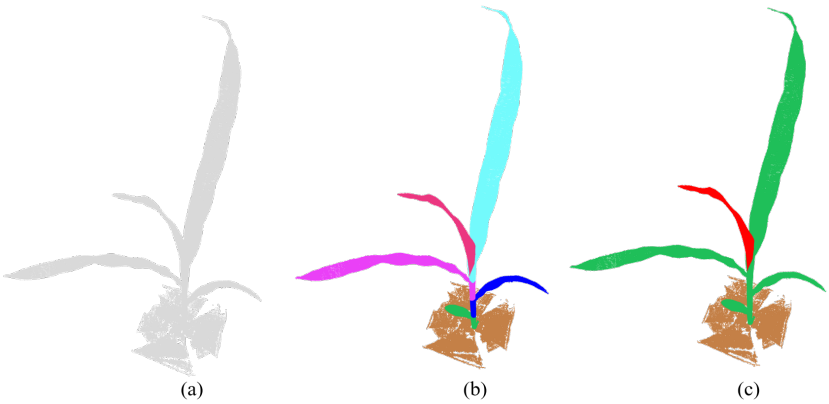

Figure 1. A 3D maize point cloud. (**a**) The raw maize plant point cloud. (**b**) Original annotation of point cloud which soil with yellow color, stem with red color, and leaf instances with purple and blue colors. (**c**) Annotation of leaves with the same label and color for semantic segmentation.

2.2. Data Pre-Processing

We manually relabeled all leaf points with the same "leaf" labels with the CloudCompare (v2.12) tool to conduct plant semantic segmentation. Figure 1 presents our relabeled semantic segmentation. In addition, although point clouds of plants were collected over 12 days, only 7 days of a maize plant's point clouds were annotated. To enhance the model's robustness and generalizability, we labeled the remaining five-day point clouds to augment the maize dataset. Specifically, we followed the Leaf Tip approach from Pheno4D to label the petioles between leafets as "stem" with branch-like structures. We eventually obtained 84 annotated point clouds of maize for model training and testing. Table 1 and Figure 2 demonstrate the average number (10^6) of points per day of maize plants measured over 12 days.

Table 1. The average number (10^6) of points of maize plants measured at each growing stage.

Growth Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	12
Maize	1.53	1.57	1.57	1.17	0.74	0.44	0.82	0.84	1.02	0.84	1.25	1.33

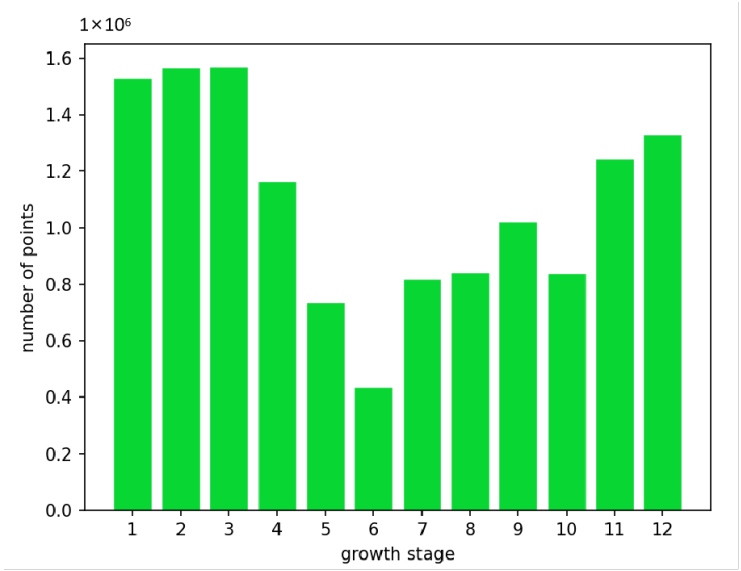

Figure 2. The average number of points measured at each growing stage.

As shown in Figure 2, the size and complexity of the maize plants' structure each day varied substantially. For the heterogeneous distribution of maize point clouds, we assigned the entire point clouds of the 2nd–6th maize plants over the 12 days for training and tested our algorithm on the point clouds of the 1st and 7th maize plants. Thus, the training set and test set contained the complete feature of point clouds of the maize at the growth phase. Table 2 presents the distribution of the dataset. The training set contains 60 point clouds, and the test set contains 24 point clouds.

Table 2. Distribution of point cloud datasets.

Category	Total	Training Set	Test Set
Maize	84	60	24

Table 3 shows the semantic distribution of points. The soil and leaf points make up the majority, while the stem has a minor proportion. The imbalance distribution of point clouds within the three classes introduces challenges to learning-based approaches, especially those concerning input point clouds.

Table 3. Average percentage of point cloud for organ class in training set and test set (%).

Category	Soil	Stem	Leaf
Training Set	50.03	5.50	44.47
Test Set	48.56	7.40	44.04

For the point-based deep learning architecture with respect to raw point data as input, which avoids the loss of spatial information due to subsampling, we used the same data strategy as PointNet [21] to process input maize point clouds. A maize point cloud was divided into blocks of size 5 cm × 5 cm, with each point being represented by its XYZ coordinates. A semi-random sampling technique was employed to sample 2048 points in each block. The final segmentation was achieved by merging the predictions from the input blocks.

2.3. Win-Former Network

Figure 3 depicts the full architecture of maize point cloud semantic segmentation. The upper is the encoder, consisting of three spherical coding blocks, while the bottom is the

decoder, which concatenates features from previous layers to produce the final feature. We transform the raw maize point cloud into a spherical representation to obtain the local neighbor areas efficiently. Firstly, we apply a Sphere Projection (SP) to all points, which would reduce the dimension used for partition, followed by a multi-layer perception (MLP) for point-wise feature transformation. We then extract the global hierarchical features by stacking three Window Transformer (WT) learning modules as a backbone for maize point cloud processing. The WT block concatenates a Sphere Window (SW) module with a neighbor searching strategy, a Cross-Window self-attention (C-SA) module, and a pooling module to aggregate local features. A stack of WT modules forms a hierarchical feature extractor. Following that, we concatenate the high-dimensional features with the features from the previous layer by expanding them to the matching windows in the next layer. After several expansions of local features in the decoder, we recover the point cloud's shape to the origin with a feature dimension of 960 (512 + 256 + 125 + 64). Finally, we pass the features through a number of fully connected layers with a softmax function to produce the final prediction.

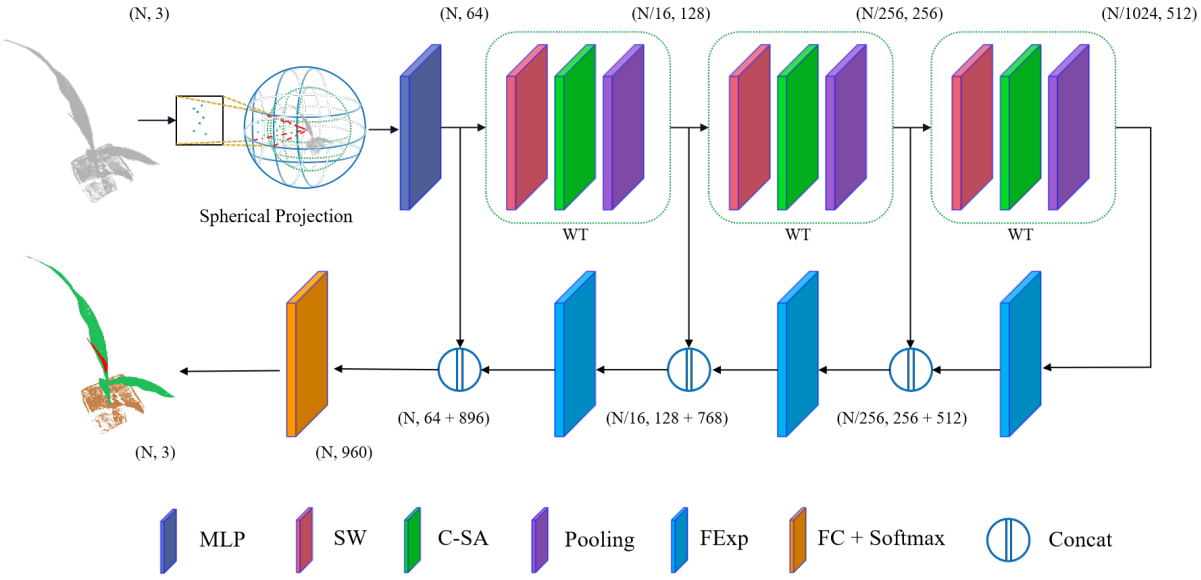

Figure 3. Win-Former architecture. The window's dots depict points which are mapped into a window from the corresponding color spheres. MLP indicates multi-layer perception. WT stands for Window Transformer. SW stands for the Sphere Window layer. C-SA is the Cross-Window self-attention. Pooling is for performing feature aggregation. FExp expands high-dimensional features to local windows. FC + Softmax stand for Fully Connected layer and Softmax function. Concat stands for concatenate operation.

2.3.1. Sphere Projection

The SP module maps 3D points onto a sphere surface. Figure 3 depicts the projection of two spheres for the sake of simplicity. Points from various facets of the same cone are mapped into a window [36]. The window's green and blue points correspond to the blue and green spheres, respectively.

Consider an input maize point cloud $P = \{p_1, \cdots p_n\} \in R^{N \times D}$, which is an unordered and irregular set including N points each with D-dimension features. In our case, N equals 2048, and D equals 3 (i.e., the point coordinates, XYZ), which may vary when the point has more features, such as RGB, normalized spatial coordinate vectors, etc. To scale the maize point clouds into a uniform coordinate space, we normalize the points as follows:

$$\hat{P} = \{\hat{p}_i \mid \hat{p}_i = p_i - p_c, \qquad p_c = \frac{1}{N}(\sum_{i=1}^{N} x_i, \sum_{i=1}^{N} y_i, \sum_{i=1}^{N} z_i); \qquad i \in N\}, \tag{1}$$

where (x_i, y_i, z_i) denote the coordinates of p_i in the Cartesian system, $\hat{p}_i$ represents the point translated from point p_i, and p_c stands for the centroid of points.

The centroid p_c of the points is converted into the origin of the spherical coordinate. Then the points are mapped from a Cartesian to a spherical coordinate system by an SP method, as follows:

$$\begin{aligned} r_i &= \sqrt{x_i^2 + y_i^2 + z_i^2} \\ \theta_i &= \arctan \frac{y_i}{x_i} \\ \varphi_i &= \arccos \frac{z_i}{r_i}, \end{aligned} \tag{2}$$

where r_i, θ_i and φ_i $(i \in N)$ denote the radius, and azimuth and elevation angles in the spherical coordinate system, respectively.

While the SP retains the majority of point features, it may change the spatial relationships among points on the sphere surface, resulting in a loss of structural information. In this way, we employ MLP to preserve the geometric connections between points. Finally, the SP module outputs features $F_c = \{f_1, \cdots f_n\} \in R^{N \times C}$, which are thus given as follows:

$$F_c = \{f_i \mid f_i = R(\hat{p}_i) \oplus M(\hat{p}_i), \qquad \hat{p}_i = r_i \oplus \theta_i \oplus \varphi_i; \qquad i \in N\}, \tag{3}$$

where C denotes the feature channel, R represents the sphere projection, and M stands for transformation of MLP. $\oplus$ denotes the concatenation operator. The angle of azimuth h_i and the elevation w_i are used to divide the features of P.

2.3.2. Window Transformer

Point clouds' irregular, unordered, and density-varied nature prevents neural networks from directly processing the 3D point clouds. Transformer is a natural fit to solve the above problem since point clouds can be treated as sets of coordinates or other features. Nevertheless, applying Transformer to a point cloud, which considers the set of points as a sequence of elements, suffers from the loss of neighboring information and computational and memory costs. Moreover, there is a significant deficiency in the hierarchy of learned features [22]. In addition, most previous networks' FPS and KNN searching algorithms also incur significant computational and memory costs, rendering them inappropriate for large-scale point clouds [32].

To solve the aforementioned problems, we construct a simple and fast hierarchical WT layer to compute L-SA with linear computational complexity [37]. The WT introduces a region-partitioning algorithm that divides the points on the sphere into multiple patches. Each patch contains points from a neighborhood. We utilize windows to arrange the splitting points along the azimuth and elevation on the sphere. The operation is invariant to the permutation point cloud. The output features F_w of a WT are defined as follows:

$$F_w = g_w(k_w(g_h(k_h(F_c)))), \tag{4}$$

where F_c is the output point feature set of the previous layer, k_h denotes a sorting function using h, g_h transform the feature with the shape of $N \times C$ to the feature with the shape of $H \times N_h \times C$. Here, $N = N_h \times H$. Similar transformations are performed by k_w and g_w.

Subsequently, we obtain a new feature set $F_w \in R^{W \times H \times N_w \times N_h \times C}$. The point cloud is divided into $W \times H$ windows, and each window contains $N_w \times N_h$ points. The SW layer ensures invariance under permutations and rotations of the point cloud by simply using sorting and reshaping methods.

The self-attention module [38] is the core component of Transformer. It constructs relations with all input features and the output of the self-attention module. As depicted in Figure 4, we adopt a L-SA algorithm on each window and gather low-dimension local features by merging patches. The local points are then translated and grouped into a

high-dimension region. We arrange the windows uniformly, covering the entire sphere to divide the points, resulting in a higher-level neighborhood.

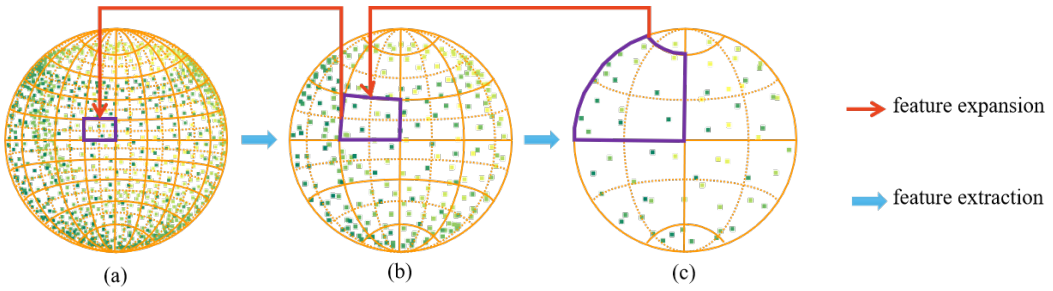

Figure 4. The Window Transformer mechanism. The WT hierarchically integrates local features inside a 4×4 spherical window to derive the global feature. The blue arrows refer to the process of feature extraction from local areas. The purple boxes depict the windows for the L-SA and pooling operations. The red arrows denote the flow of feature expansion from high-dimension space to low-dimension space. The depiction simplifies the structure to present the module principle.

The KNN with Euclidean distance, employed by most previous works as the local neighborhood selection strategy, has computational complexity concerning the points' number. For example, consider a window with a shape of (h, w) consisting of $h \times w$ points; the global self-attention's computational complexity on a sphere of $H \times W$ is quadratic $H \times W$ points, calculated as follows:

$$\Omega(Self\_Atten) = 4HWC^2 + 2(HW)^2C, \tag{5}$$

The computational complexity of our Window-based L-SA performed on the same sphere is linear to the number of points, calculated as follows:

$$\Omega(Win\_Self\_Atten) = 4HWC^2 + 2(HW)(hw)C, \tag{6}$$

Our Window-based self-attention layer is faster than global self-attention and is scalable to point number and data dimensions. Additionally, the Window-based mechanism reduces the number of required patches from N to $\frac{N}{K}$ with a K window size. In our case, it is reduced from 2048 points to 128 through a window (in our experiment, the window size is 4×4).

2.3.3. Cross-Window Self-Attention

We apply a C-SA algorithm to capture connections across neighboring windows. As illustrated in Figure 5, after L-SA is calculated, windows are turned along the azimuth and elevation angles by $(h/2, w/2)$ to calculate the self-attention (the h and w are the window size). In our case, the window size is 4×4 and rotating by $(2, 2)$. Our C-SA method establishes relations among neighboring windows and boosts the model's performance.

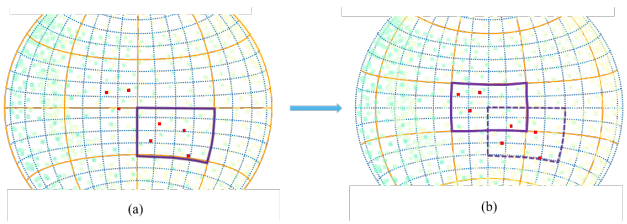

Figure 5. Cross-Window self-attention. In (**a**), L-SA is computed in the windows, and then the entire window moves along azimuth and elevation in (**b**). The yellow box depicts a sphere window. The blue box indicates a patch containing some points. The window that matches the window in (**a**) is represented by the purple dashed box in (**b**), and the window after rotation is indicated by the purple solid box in (**b**).

2.4. Experiments and Setting

For a fair comparison, all networks were tested on an Nvidia Geforce RTX 3090 GPU (Nvidia, Santa Clara, CA, USA). The networks were tested on PyTorch [39] with a batch size 32 for 200 epochs and trained using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of 0.0001 for training. The initial learning rate is 0.001 and adjusted at every epoch using the cosine-annealing strategy. We augmented the dataset by applying a random translation by the range of $[-0.25, 0.25]$ and a random anisotropic scaling by the range of $[0.59, 1.4]$ during training. In contrast, we do not augment data during the testing phase.

2.5. Evaluation Metrics

We used the Intersection over Union (IoU) and mean class Intersection over Union (mIoU) as evaluation metrics for the maize point cloud understanding tasks. TP_c (true positives), FP_c (false positives), and FN_c (false negatives) denote the predictions of category C, respectively, where $C \in \{soil, stem, leaf\}$ indicates the organic class of a maize point cloud.

The IoU for each class is calculated as follows:

$$IoU_c = \frac{TP_c}{TP_c + FN_c + FP_c}. \tag{7}$$

The mIoU over all classes is calculated as follows:

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c}. \tag{8}$$

3. Results and Discussion

We evaluated our proposed Win-Former network with three famous networks, including PointNet [21], PointNet++ [22], and DGCNN [40] on the maize point cloud dataset for segmentation tasks. We then conducted 3D semantic segmentation on shapeNet55 [41] to validate the robustness and generalizability of the Win-Former model. We further performed a visualization of our approach. Finally, we conducted an ablation study to validate the network.

3.1. Results on Maize Dataset

Table 4 demonstrates the superiority of the Win-Former over the PiontNet, PiontNet, and DGCNN models on the maize dataset. Our Win-Former consistently outperforms representative models in all three categories in balancing accuracy and inference speed. Specifically, all methods obtain relatively good IoU, which is more than 98% in the ground category, and rank second highest in IoU, which is higher than 83% in the leaf category. However, PointNet struggles with the relative mall stem class and only achieves a 4.25% IoU in the stem category, meaning it hardly identifies the stem. PointNet++ and DGCNN achieve a relatively higher IoU (<40%) in the stem category, meaning these two methods could not distinguish stems from leaves. Our Win-Former achieves a 58.43% IoU in the stem category, outperforming other methods by a large margin (>19%). Compared to PiontNet, PiontNet++, and DGCNN, our Win-Former makes a 25.74%, 10.61%, and 8.86% mIoU improvement, respectively. The high performance of our model is due to the fact that local point information is preserved and enhanced by using transformer operations on points within each window and concatenating them by the cross-windows strategy.

Table 4. The mIoU and IoU of each class on maize dataset.

Method	Category	IoU	mIoU	Latency
PointNet [21]	Soil	98.25	61.97	39
	Stem	4.25		
	Leaf	83.42		
PointNet++ [22]	Soil	99.3	74.6	49
	Stem	36.67		
	Leaf	87.85		
DGCNN [40]	Soil	99.45	76.06	64
	Stem	39.3		
	Leaf	89.45		
Win-Former	Soil	99.67	83.45	31
	Stem	58.43		
	Leaf	92.25		

Moreover, our Win-Former also has the lowest latency of 31 s. The main reason is that the windowing mechanism of our Win-Former searches for points only through azimuth and elevation angles, rather than the XYZ dimension employed by other methods. The windows establishing a geometric relationship of local point features play a role like the kernel of traditional convolutional neural networks (CNN). Our Win-Former can be applied in near-real-time environments due to its high efficiency. In contrast, the above point-based methods extract point representations by pooling point features with respect to the high density of input point clouds.

Figure 6 is the visualization results of semantic segmentation on maize dataset. Figure 6 illustrates that all networks identify the leaf and soil well. However, PointNet completely misses the stems and recognizes them as leaf categories. Due to extracting features with a hierarchical SA module, PointNet++ identifies parts of the stems while misidentifying the leaf as the stem in row b. Due to the dynamic edge convolution in each layer, DGCNN improves recognizing the entire stem or a portion of it, as seen in rows b and c. Nevertheless, a large portion of the leaf in row a, the boundary of the leaf in row b, and the transition from stem to leaf in row d are misclassified as stems. In contrast, our Win-Former network outputs smoother predictions and is robust to all maize plants. However, our Win-Former is only able to identify a portion of the stem.

Discussion. Thanks to the power of extracting relationships within the neighbor points of the Transformer, our proposed model outperforms several famous point cloud learning methods. PointNet, which depends only on an MLP operator, suffers from low identification results, especially for small category objects, i.e., the stem. The main reason is a lack of learning relationships between different points. As an updated version, PointNet++ introduces a local feature learning module to construct a hierarchical structure and obtains a better recognition. DGCNN introduces a strong graph structure to perform Edge Convolution, which can capture more detailed information, resulting in more accurate recognition results. However, the complicated structure also takes more time than others, which would impede its application to large-scale data. Our method benefits from the self-attention mechanism, which can compute an attention score according to the point features within a neighboring area, and improves the mIoU of each category to a new level. In addition, our method does not consume more time than other methods due to the sphere projecting operation reducing the dimensions of raw data. The self-attention mechanism helps to capture detailed information and therefore performs well on objects with small proportions.

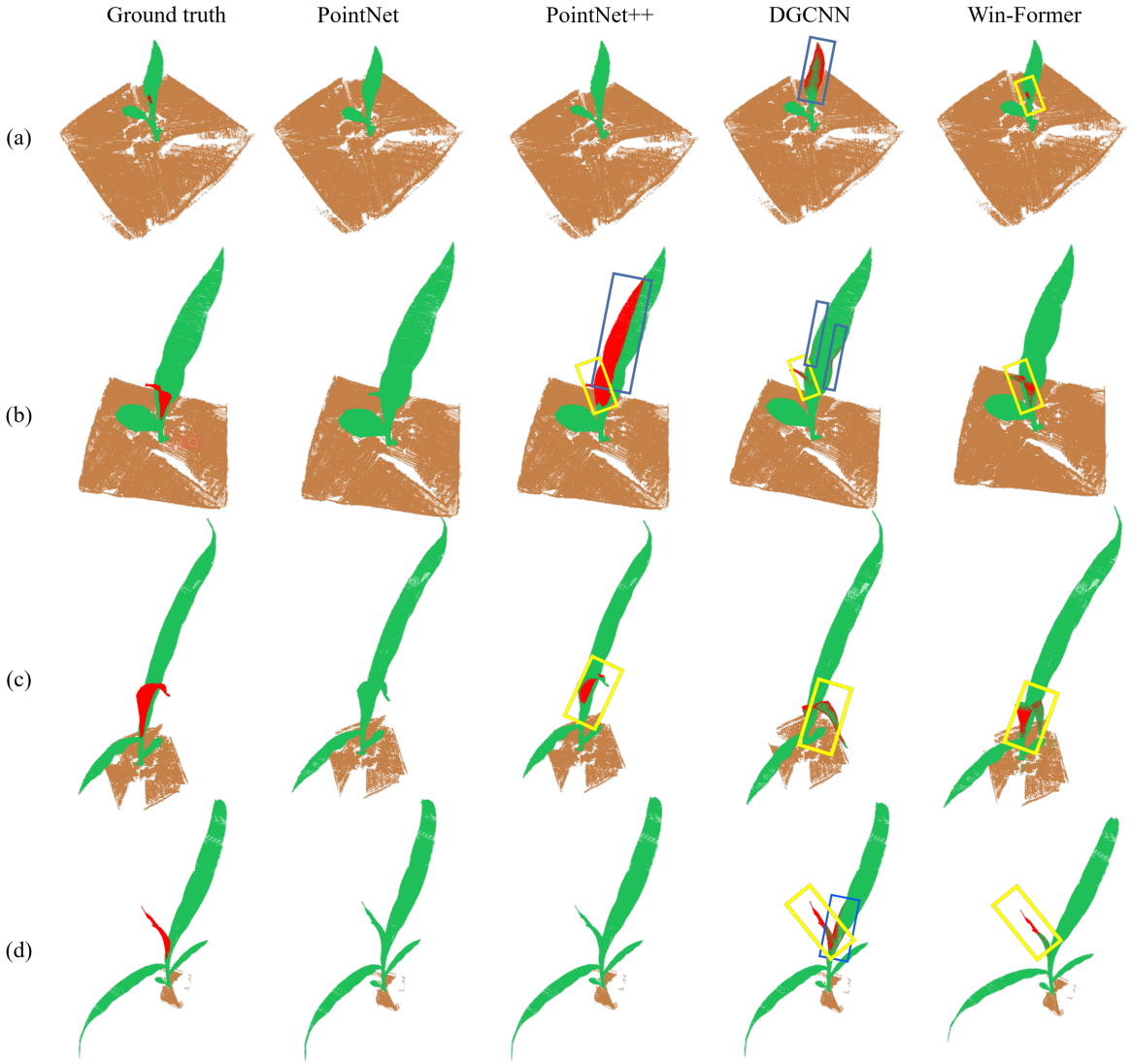

Figure 6. Visualization for maize point cloud segmentation presented in the same camera viewpoint. Rows a to d depict maize plants in four stages of growth. Yellow rectangles indicate detected stems, and blue rectangles indicate misidentified points.

3.2. Results on ShapeNet

To further evaluate the generalizability and robustness of the Win-Former model, we performed the method on the ShapeNet55 [41] dataset for the part segmentation benchmark. In the dataset, there are 16,880 3D point clouds, with 14,006 models in the training set and 2874 in the testing set. The dataset has 16 classes annotated with 50 parts. We then compare our Win-Former with a representative set of previous works, including point-based and Transformer-based networks. Table 5 demonstrates the per-category and mean part IoU (%). While the data are challenging, we can see that our predictions are reasonable. Our Win-Former obtains competitive results compared with the SOTA networks in part segmentation. The 85.4% mIoU indicates that our Win-Former network can generate accurate features for fine-grained learning.

A particularly appealing phenomenon is that our proposed method produces a subset of results that are larger in mIoU than others, such as ear phone, laptop, and mug. While there is a slight gap between Win-Former and PCT, we think overlapping and shaded points (i.e., aero and lamp) caused semantic ambiguity when projecting on a spherical surface.

Figure 7 shows the visualization of part segmentation in the ShapeNet55 dataset. We can see that Win-Former's prediction is accurate and close to the ground truth. Win-Former

performs well in capturing the structural details, such as the legs of the chair and wheels of the car.

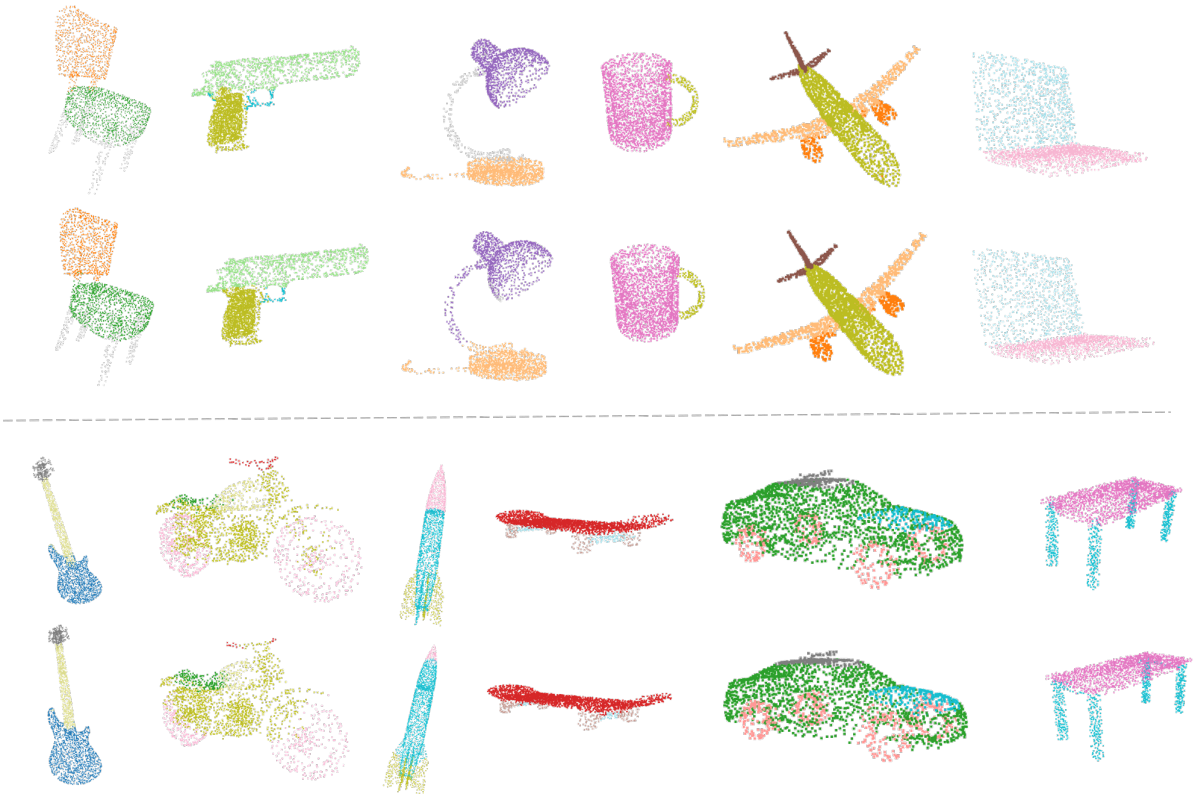

Figure 7. Visualization of part segmentation results on ShapeNet. The ground truth is shown in the top row, and the results of our Win-Former are displayed in the bottom row.

Discussion. As illustrated in Table 5, our method performs better than previous SOTA methods, except for PCT. Using Transformer as the core module, PCT achieves the best results of segmentation and outperforms other methods in mIoU. However, PCT builds the hierarchical feature extraction by employing KNN and the ball query strategy, which have a computation complexity of $O(n^2)$. This limits its expansion to large-scale point data. Our method consider both the effectiveness and efficiency by adopting projection to descent dimension and window-based Transformer to capture detailed information. Additionally, our Win-Former obtains the best results of IoU in the laptop, mug, and table categories. Since sphere projection would bring in some ambiguities due to the obscured parts, our proposed method suffers from recognizing complicated categories, such as the car and motor. The next stage of our work is to solve the shelter issue and generalize the method to more complicated plants.

3.3. Ablation Study

We perform contrast experiments to evaluate how much the window size (h, w) and how many C-SA layers are needed to achieve satisfactory performance. We also investigate the influences of the type of pooling operators which are employed to aggregate the local features within the sphere windows. We evaluate all ablation studies on the maize plant point cloud semantic segmentation, and report the mIoU results of maize plant no. 7.

Table 5. The mIoU and IoU of each part on the ShapeNet dataset. Results of other methods are quoted from the cited papers.

Method	mIoU	Aero	Bag	Cap	Car	Chair	Ear Phone	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate Board	Table
PointNet [21]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
SO-Net [42]	84.9	82.8	77.8	88.0	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
PointNet++ [22]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
PointCNN [43]	86.1	84.1	86.5	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.2	84.2	64.2	80.0	83.0
PCNN [44]	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
DGCNN [40]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
PCT [32]	86.4	85.0	82.4	89.0	81.2	91.9	71.5	91.3	88.1	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7
Win-Former	85.4	82.2	79.2	83.1	74.3	91.7	74.7	91.7	86.0	80.6	97.7	55.8	96.6	83.3	53.0	74.5	83.9

3.3.1. Sphere Window Size

Table 6 shows that Win-Former achieves the best performance at $(4, 4)$ for the window size. Since the window determines the local region where self-attention is performed, a suitable scale is required for the Transformer blocks. On the one hand, a small window which performs self-attention computation efficiently would have insufficient neighboring points for learning features. On the other hand, a large window has more neighboring points to extract abundant information, but it also introduces more computational consumption.

Table 6. Semantic segmentation results on maize datasets.

Window Size	Layers	Initial Points	mIoU
2×2	4	2	81.28
3×3	3	2	82.25
4×4	2	8	83.45
8×8	1	32	79.23

3.3.2. Cross-Attention

In Figure 8, we report the L-SA and C-SA mechanisms in each layer. As illustrated in Figure 8, the performance of cross-attention within a layer is 1% better than local attention, suggesting that inserting a cross-attention module helps. When applying C-SA three times, the mIoU reaches a peak, while continuing to add the C-SA module in the layer decreases the performance of the model. This situation suggests that an appropriate number of C-SA could learn adequate information between different windows.

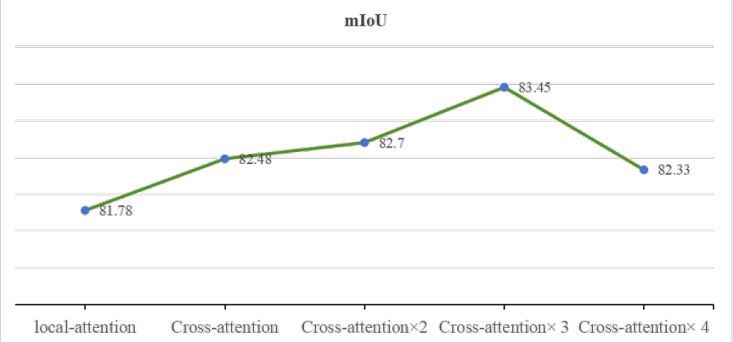

Figure 8. Performance (average mIoU) of different attention mechanisms.

3.3.3. Type of Pooling

The ablation results of the pooling type are shown in Table 7. The Max-Pool mechanism obtains the best results. An interesting scenario is that the performance drops compared with Max-Pool even if we concatenate the max features and the average feature together. This suggests that Max-Pool could learn the most significant features in the neighboring area.

Table 7. Type of pooling methods. Max-Pool and Avg-Pool denote the max pooling and average pooling, respectively. The Max-Avg is the concatenation of the results of Max-Pool and Avg-Pool. Con-Pool means the concatenation of features in a window.

Pooling Type	mIou
Max-Pool	83.45
Avg-Pool	81.9
Max-Avg	82.3
Con-Pool	81.6

4. Conclusions

Semantic segmentation of plant organs is a critical component of high-throughput plant phenotyping systems. We propose a Win-Former network to segment the 3D maize point cloud. We project the maize points onto a sphere to reduce the points' dimensionality. We also introduce Window Transformer and Cross-Window attention algorithms to extract hierarchical features. Our Win-Former achieves the highest IoU of 99.67% in the soil, 58.43% in the stem, and 92.25% in the leaf category. In addition, the model receives the highest mIoU of 83.45% and the lowest latency of 31 s. Moreover, we also achieve competitive results on ShapeNet for the shape segmentation task.

Our Win-Former proposes Sphere Projection and Window Transformer to gather local features, but it is still weak in directly capturing fine-grained contexts. In the future, we will investigate feature extraction of semantic segmentation for plant phenotyping analysis.

Author Contributions: Conceptualization, Y.S., H.Y. and X.G.; methodology, Y.S. and X.G.; software, Y.S. and X.G.; validation, X.G. and H.Y.; formal analysis, H.Y.; investigation, X.G.; resources, X.G.;writing—original draft preparation, Y.S. and X.G.; writing—review and editing, Y.S. and H.Y.; visualization, H.Y. and X.G.; supervision, H.Y.; project administration, H.Y., funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Program of Shanxi Province (No. 202203021222175) and the Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (No. 2022L086).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank David Schunck et al. for the Pheno4D dataset.

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

References

1. Ranum, P.; Peña-Rosas, J.P.; Garcia-Casal, M.N. Global maize production, utilization, and consumption. *Ann. N. Y. Acad. Sci.* **2014**, *1312*, 105–112. [CrossRef] [PubMed]
2. Ngoune Tandzi, L.; Mutengwa, C.S. Estimation of Maize (*Zea mays* L.) Yield Per Harvest Area: Appropriate Methods. *Agronomy* **2020**, *10*, 29 . [CrossRef]
3. Revilla, P.; Anibas, C.M.; Tracy, W.F. Sweet Corn Research around the World 2015-2020. *Agronomy* **2021**, *11*, 534. [CrossRef]
4. Araus, J.L.; Cairns, J.E. Field high-throughput phenotyping: The new crop breeding frontier. *Trends Plant Sci.* **2014**, *19*, 52–61. [CrossRef] [PubMed]
5. Chaivivatrakul, S.; Tang, L.; Dailey, M.N.; Nakarmi, A.D. Automatic morphological trait characterization for corn plants via 3D holographic reconstruction. *Comput. Electron. Agric.* **2014**, *109*, 109–123. .: 10.1016/j.compag.2014.09.005. [CrossRef]
6. Zhou, J.; Tardieu, F.; Pridmore, T.; Doonan, J.; Reynolds, D.; Hall, N.; Griffiths, S.; Cheng, T.; Zhu, Y.; Jiang, D.; et al. Plant phenomics: History present status and challenges. *J. Nanjing Agric. Univ.* **2018**, *41*, 9.
7. Huichun, Z.; Hongpin, Z.; Jiaqian, Z.; Yufen, G.; Yangxian, L. Research Progress and Prospect in Plant Phenotyping Platform and Image Analysis Technology. *Trans. Chin. Soc. Agric. Mach.* **2020**, *51*, 17.
8. Vázquez-Arellano, M.; Reiser, D.; Paraforos, D.S.; Garrido-Izard, M.; Burce, M.E.C.; Griepentrog, H.W. 3-D reconstruction of maize plants using a time-of-flight camera. *Comput. Electron. Agr.* **2018**, *145*, 235–247. [CrossRef]
9. Forero, M.G.; Murcia, H.F.; Méndez, D.; Betancourt-Lozano, J. LiDAR Platform for Acquisition of 3D Plant Phenotyping Database. *Plants* **2022**, *11*, 2199. [CrossRef]
10. Sun, G.; Wang, X. Three-Dimensional Point Cloud Reconstruction and Morphology Measurement Method for Greenhouse Plants Based on the Kinect Sensor Self-Calibration. *Agronomy* **2019**, *9*, 596. [CrossRef]
11. Zhang, Y.; Sun, H.; Zhang, F.; Zhang, B.; Tao, S.; Li, H.; Qi, K.; Zhang, S.; Ninomiya, S.; Mu, Y. Real-Time Localization and Colorful Three-Dimensional Mapping of Orchards Based on Multi-Sensor Fusion Using Extended Kalman Filter. *Agronomy* **2023**, *13*, 2158. [CrossRef]
12. Yuan, Z.; Song, X.; Bai, L.; Wang, Z.; Ouyang, W. Temporal-Channel Transformer for 3D Lidar-Based Video Object Detection for Autonomous Driving. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 2068–2078. [CrossRef]
13. Wang, Q.; Kim, M.K. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Adv. Eng. Inform.* **2019**, *39*, 306–319.
14. Han, L.; Zheng, T.; Zhu, Y.; Xu, L.; Fang, L. Live Semantic 3D Perception for Immersive Augmented Reality. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 2012–2022. [CrossRef] [PubMed]

15. Yan, Y.; Zhang, B.; Zhou, J.; Zhang, Y.; Liu, X. Real-Time Localization and Mapping Utilizing Multi-Sensor Fusion and Visual–IMU–Wheel Odometry for Agricultural Robots in Unstructured, Dynamic and GPS-Denied Greenhouse Environments. *Agronomy* **2022**, *12*, 1740. [CrossRef]
16. 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]
17. Elnashef, B.; Filin, S.; Lati, R.N. Tensor-based classification and segmentation of three-dimensional point clouds for organ-level plant phenotyping and growth analysis. *Comput. Electron. Agric.* **2019**, *156*, 51–61. [CrossRef]
18. Wang, Y.; Hu, S.; Ren, H.; Yang, W.; Zhai, R. 3DPhenoMVS: A Low-Cost 3D Tomato Phenotyping Pipeline Using 3D Reconstruction Point Cloud Based on Multiview Images. *Agronomy* **2022**, *12*, 1865. [CrossRef]
19. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
20. Wang, P.S.; Liu, Y.; Guo, Y.X.; Sun, C.Y.; Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–11. [CrossRef]
21. 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 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
22. Qi, C.R.; Li, Y.; Hao, S.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv* **2017**, arXiv:1706.02413.
23. Turgut, K.; Dutagaci, H.; Galopin, G.; Rousseau, D. Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods. *PLant Methods* **2022**, *18*, 20. [CrossRef] [PubMed]
24. Li, Y.; Wen, W.; Miao, T.; Wu, S.; Yu, Z.; Wang, X.; Guo, X.; Zhao, C. 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]
25. 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. [CrossRef] [PubMed]
26. Guo, X.; Sun, Y.; Yang, H. FF-Net: Feature-Fusion-Based Network for Semantic Segmentation of 3D Plant Point Cloud. *Plants* **2023**, *12*, 1867. [CrossRef] [PubMed]
27. Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; Chen, L.C. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In Proceedings of the Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5463–5474. [CrossRef]
28. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. In Proceedings of the SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, Online, 6–12 December 2021.
29. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *arXiv* **2021**, arXiv:2102.12122v2.
30. Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Tomizuka, M.; Keutzer, K.; Vajda, P. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. *arXiv* **2020**, arXiv:2006.03677.
31. Yu, J.; Zhang, C.; Wang, H.; Zhang, D.; Song, Y.; Xiang, T.; Liu, D.; Cai, W. 3D Medical Point Transformer: Introducing Convolution to Attention Networks for Medical Point Cloud Analysis. *arXiv* **2021**, arXiv:2112.04863.
32. 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]
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
34. Cui, Y.; Fang, Z.; Shan, J.; Gu, Z.; Zhou, S. 3D Object Tracking with Transformer. *arXiv* **2021**, arXiv:2110.14921.
35. Schunck, D.; Magistri, F.; Rosu, R.A.; Cornelißen, A.; Chebrolu, N.; Paulus, S.; Léon, J.; Behnke, S.; Stachniss, C.; Kuhlmann, H.; et al. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. *PloS ONE* **2021**, *16*, e0256340. [CrossRef]
36. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
37. Guo, X.; Sun, Y.; Zhao, R.; Kuang, L.; Han, X. SWPT: Spherical Window-Based Point Cloud Transformer. In Proceedings of the Computer Vision—ACCV 2022, Macao, China, 4–8 December 2022; pp. 396–412. [CrossRef]
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–6 December 2017; Volume 30.
39. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada, 8–14 December 2019.
40. Simonovsky, M.; Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3693–3702. [CrossRef]
41. 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]

42. Li, J.; Chen, B.M.; Lee, G.H. So-net: Self-organizing network for point cloud analysis. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9397–9406.
43. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On X-Transformed Points. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
44. Atzmon, M.; Maron, H.; Lipman, Y. Point convolutional neural networks by extension operators. *arXiv* **2018**, arXiv:1803.10091.