Computers and Electronics in Agriculture 221 (2024) 108922

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

PlantSegNet: 3D point cloud instance segmentation of nearby plant organs with identical semantics

Ariyan Zarei [a,*], Bosheng Li [b], James C. Schnable [c], Eric Lyons [d], Duke Pauli [d], Kobus Barnard [a], Bedrich Benes [b]

[a] Department of Computer Science, The University of Arizona, Tucson, 85721, AZ, USA
[b] Department of Computer Science, Purdue University, West Lafayette, 47906, IN, USA
[c] Center for Plant Science Innovation and Department of Agronomy and Horticulture, University of Nebraska–Lincoln, Lincoln, 68588, NE, USA
[d] School of Plant Science, The University of Arizona, Tucson, 85721, AZ, USA

ARTICLE INFO

Keywords:
Phenotyping
Digital twins
Point clouds
Procedural modeling
Plant geometry
Sorghum

ABSTRACT

In this study, we introduce PlantSegNet, a novel neural network model for instance segmentation of nearby objects with similar geometric structures. Our work addresses the challenges of instance segmentation of plant point clouds, including the difficulty of annotating and labeling point clouds, the loss of local structural information in neural network components, and the generation of large numbers of incorrect small clusters due to poor choices of the loss function. One of the key contributions of our approach is a digital twin of sorghum, i.e., a procedural sorghum model, which was used to generate point clouds of sorghum fields. This allowed us to create a large-scale, annotated, synthetic dataset of sorghum plants that we used to train our PlantSegNet model. We demonstrated the effectiveness of our method in segmenting instances of sorghum leaves grown in outdoor field settings. To the best of our knowledge, this is the first study to address this specific instance segmentation problem for plants grown in such a setting. We compared our proposed method with other state-of-the-art methods for indoor settings, including SGPN and TreePartNet, on both synthetic and real data. Our results show that PlantSegNet outperforms these methods regarding accuracy, robustness, and efficiency.

1. Introduction

1.1. Problem definition

Automated acquisition and analysis of plant phenotypes are critical in studying plant biology and genetics (Zarei et al., 2022). 3D plant structure and characteristics are among the most important phenotypes and, simultaneously, one of the most challenging ones to acquire and analyze (Li et al., 2022b). Point clouds are one of the most common data acquired, and their analysis is an outstanding problem. Instance segmentation assigns points to different plant organs, such as leaves or stems. There have been significant efforts towards instance segmentation of plants, but it remains challenging for four reasons. Firstly, labeling point clouds is arduous and complex, leading to a need for large-scale labeled datasets for segmenting various plant organs. Secondly, neural network components used in prior works tend to lose local structural information, resulting in poor segmentation outcomes. Thirdly, unsuitable choices of loss functions in feature-space-learning methods can produce an excessive number of incorrect small clusters. Finally, most previous studies have focused on plants grown separately in pots or transplanted during imaging, and no method has been proposed for segmenting point clouds of plants grown in outdoor field settings. To overcome these challenges, we introduce PlantSegNet, a neural network model designed for instance segmentation of nearby objects with identical semantics, specifically for segmenting instances of sorghum leaves grown in outdoor field settings. In our study, we train our proposed neural network structure, i.e., PlantSegNet, on an extensive synthetic dataset based on realistic computer 3D plant models (digital twins).

1.2. Related work

Traditional methods for obtaining structural and morphological plant phenotypes are labor-intensive and time-consuming, making the process slow and not scalable (Li et al., 2022b; Luo et al., 2022). With recent advancements in AI-based computer vision techniques, high-throughput phenotype extraction, especially 3D plant structure

* Corresponding author.
E-mail address: ariyanzarei@arizona.edu (A. Zarei).

https://doi.org/10.1016/j.compag.2024.108922
Received 20 June 2023; Received in revised form 26 March 2024; Accepted 5 April 2024
Available online 17 April 2024

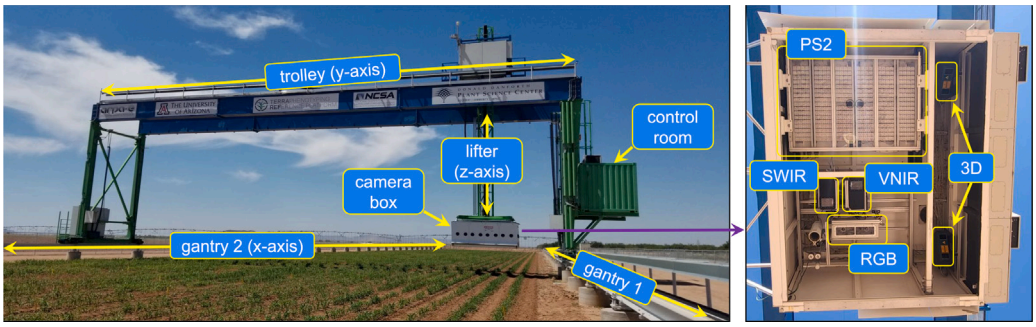

Fig. 1. The gantry system is located in Maricopa County, Arizona. (left) The gantry moves on two rails (gantry 1 and 2 x-axis). The camera box moves horizontally on the trolley (y-axis) and vertically on a lifter (z-axis), providing three degrees of freedom. Different crops are grown and scanned daily using various high-resolution sensors and scanners. The right image shows a view into the camera box with sensors providing RGB color images, point clouds (3D), hyperspectral images (VNIR and SWIR), and images in the range of 690–730 nm (PS2). The setup also includes an MWIR/LWIR thermal imager (FLIR) mounted outside of the camera box (not shown).

analysis, has gained significant attention in plant and computer science communities.

Image-based analysis of plant phenotypes has been widely studied, and these methods have limitations in measuring plant structural characteristics due to the single view and plant organ occlusion (Luo et al., 2022) or full reconstruction (Li et al., 2021). With the introduction of 3D laser scanners and improvements in 3D computer vision techniques, researchers have shifted their focus towards 3D plant structure extraction and analysis using point clouds. Segmentation of individual plant organs is a key component of this analysis and enables the computation of 3D structural characteristics such as leaf dimensions, curvature, and angles (Li et al., 2022b).

While the fundamentals of 3D point cloud analysis, including instance and semantic segmentation, have been studied, there have been fewer efforts in segmenting plant point clouds that require separating different instances of plant organs in close proximity. Our motivating context is assessing how different water-stress treatments of 9,000 individual Sorghum plants of different genotypes grown in outdoor fields affect their structural characteristics. This requires accurately segmenting different instances of leaves in the plants. We obtained 3D point cloud data using two custom-built laser scanners mounted on a specialized, ground-based gantry system that scans a two-acre field throughout the day and night (see Fig. 1).

Several research studies have investigated instance segmentation of plant organs through conventional machine learning techniques, such as geometry-based (Wu et al., 2013) or octree-based methods (Duan et al., 2016). However, these methods are hindered by manual parameter tuning, which is both laborious and time-consuming, making them impractical for large-scale applications. Meanwhile, applying neural networks and deep learning for this task is limited and challenging due to the significant amount of labeled data that these methods require. Labeling vast quantities of 3D point cloud datasets is tedious and time-consuming, mainly due to the absence of a user-friendly 3D point cloud annotation tool. Therefore, the few existing neural network approaches for segmenting plant organs are either weakly supervised, like in Luo et al. (2022), or are solely trained on synthetic datasets, such as TreePartNet (Liu et al., 2021b).

Generally, 3D geometric structures can be stored in various representations, such as 3D point clouds, depth images, volumetric grids, and meshes. Deep learning methods for 3D data can be categorized based on the input data representation and the feature extraction strategies utilized to enable appropriate inferences. Point clouds are prevalent among the different 3D data representations because they can be readily captured by accessible technology. However, compared to other data modalities such as images, videos, and text, analyzing objects within 3D point clouds using deep learning methods is more challenging due to its unstructured nature (Guo et al., 2020). Working with point clouds has been challenging because isolated points do not capture the topological properties of the underlying geometry. Although many efforts have addressed this challenge, the whole 3D geometry reconstruction from point clouds is still an important open problem. One fundamental problem exacerbating this issue is the lack of labeled data.

Another class of related methods includes unsupervised or self-supervised learning for point cloud pre-training. The point cloud pre-training (Yamada et al., 2022) generates a point cloud fractal database inspired by fractal self-similarity across scales. The PointVST (Zhang and Hou, 2023) uses the duality between images and point clouds to learn features in the regular image domain as opposed to irregular images, and the PointMCD (Zhang et al., 2023a) introduces the visibility-aware feature projection to aggregated point-wise embeddings into view-specific geometric descriptors. Zhang et al. (2023b) introduced the 2D point geometry image to flatten the point clouds. Many methods used masking and transformers (e.g., Pang et al., 2022) to improve learning. These methods could alleviate the need for synthetic data.

Deep learning methods for 3D data can be classified into three main categories for point cloud classification and segmentation: Projection-Based, Volumetric-Based, and Point-Based. Projection-based methods, also known as multi-view-based methods, project the 3D point cloud into multiple views and learn view-wise features for classification or segmentation. There have been many efforts to address the challenges of this class of methods dealing with information loss due to the projection and properly extracting a global feature vector by fusing the view-wise features. Examples of these methods include but are not limited to MVCNN (Su et al., 2015), MHBN (Yu et al., 2018), and View-GCN (Wei et al., 2020). Volumetric-based methods, on the other hand, convert the point clouds into a 3D grid of voxels and use 3D convolution neural networks (CNN) for classification or segmentation. Although VoxNet (Maturana and Scherer, 2015) and 3DShapeNets (Wu et al., 2015) have shown promising results, they are computationally and memory inefficient due to the voxelized representation, and they do not work with stochastic geometries. Additionally, Riegler et al. (2017), and Wang et al. (2017) employed Octrees to address the fixed resolution of the voxel-based methods.

Point-based methods work with pure, unordered point clouds and can be categorized into groups based on their approach to learning local structural relations between the points. One of the earliest efforts in classifying and semantically segmenting 3D objects in point clouds is the PointNet method (Qi et al., 2017a). They discussed the limitations of volumetric-based methods to analyze 3D data and introduced a neural network architecture for classifying and semantically segmenting 3D point clouds. PointNet uses a small alignment network at the beginning of its architecture to transform the input points into a canonical space, which makes it invariant to rigid transformations. Following that, PointNet learns per-point features using multiple shared MLPs. Finally, to make the model permutation invariant and address the unordered point cloud format, a symmetry function (max pooling) is applied to the per-point features to obtain a global feature for the whole point cloud.

The global feature is fed into a dense MLP network for classification to obtain the final class. The global feature is concatenated with the per-point features for semantic segmentation and passed through two more shared MLP networks to obtain per-point class labels.

PointNet does not consider the local relations between neighboring points, as each point learns the features individually. To tackle this limitation, Qi et al. (2017b) introduced a hierarchical method called PointNet++. Unlike PointNet, instead of max pooling the per-point features, which in turn results in losing local structural relations, PointNet++ proposes a hierarchical grouping of points using farthest-point sampling. At each level of this hierarchy, points are grouped using a distance-based approach to form neighborhood balls with overlaps. The points in each group are then processed using a simplified version of PointNet to extract local features. This process is repeated recursively until a global feature vector is generated. For classification, the global feature representation is passed through a set of fully connected layers to determine the class scores. At the same time, for semantic segmentation, a hierarchical propagation strategy is used to interpolate the per-point labels based on their distance back to the original input. By adopting this approach, PointNet++ successfully records and uses local structural relations between neighboring points, which are not taken into account in PointNet.

The PointNet++ method has limitations in capturing local structural information, which led to the introduction of graph-based approaches, a subgroup of point-based methods. Dynamic Graph Convolution Neural Networks (DGCNN) (Wang et al., 2019b), a widely used graph-based neural network for classification and semantic segmentation, introduced EdgeConv, a novel idea for simulating the convolution operation on point clouds. EdgeConv creates a directed graph using KNN to represent the local structure and vicinity of the points. An "edge feature" function calculates features for each edge of this graph using the geometrical relations between the points rather than raw values of the input space. The parameters of this non-linear function are learned using a shared MLP. The edge features of each point are then merged using a symmetry function like max pooling, making the whole process similar to convolution. The DGCNN structure is a stack of EdgeConv layers, with dynamically updating neighborhood graphs, max pooling operations, and a spatial transform network initially, similar to PointNet. EdgeConv's ability to capture local structural features of point clouds is also used in our work. Several methods have been introduced to learn point clouds using graph neural networks. Grid-GCN (Xu et al., 2020) uses a novel coverage-aware grid query on regular grids to improve spatial coverage, and Zhou et al. (2021) introduced a method that varies the size of the point cloud kernels according to their dynamically learned features. The RegGeoNet (Zhang et al., 2022) is suitable for large-scale point clouds as it parameterizes an unstructured point set into a regular deep geometry image. The transformer has been applied to learning point clouds in Zhao et al. (2021) and extended to point cloud classification in Lai et al. (2022).

Our work focuses primarily on instance segmentation, which is more challenging than semantic segmentation because it requires a more precise understanding of the points. Two main approaches exist, for instance the segmentation of objects in 3D point clouds: proposal-based and proposal-free methods. In proposal-based methods, an object detection module first obtains or proposes the boundaries of the objects in the scene, followed by another module that generates instance masks for each proposed bounding box (Hou et al., 2019; Yi et al., 2019; Yang et al., 2019). Conversely, proposal-free methods do not rely on object detection but directly work on discriminative feature learning and point grouping. One of the pioneering works in this field is the Similarity Group Proposal Network (SGPN) by Wang et al. (2018), which reduces the problem of instance segmentation to learning a discriminative embedding space. SGPN learns a feature embedding for all the points using a feed-forward neural network similar to PointNet. It generates a similarity matrix between each pair of points by taking the L_2 norm of their feature vectors in the embedding space. A double-hinge loss function is applied to the similarity matrix to enforce the distance between points belonging to the same instance to be below a threshold. Besides SGPN, there have been some other efforts in doing proposed free instance segmentation, most of which through learning a feature embedding for the points (Wang et al., 2019a; Pham et al., 2019; Liang et al., 2020). We adopt the idea of learning a discriminative feature space in our work.

The instance segmentation methods mentioned in the previous paragraph primarily focus on general 3D objects and commonly used benchmark datasets such as S3DIS (Armeni et al., 2016) and PartNet (Mo et al., 2019a). However, our specific application is the segmentation of instances of nearby objects that share the same semantics and similar shapes, specifically segmenting different instances of leaves in sorghum plants. Previous research has also explored this topic in recent years. For example, Li et al. (2022b) presented DeepSeg3DMaize, a high-throughput plant point cloud instance segmentation method. Their dataset was generated using the multiview stereo technique (Wu et al., 2020) and includes point clouds of individual plants with a fixed number of leaves (3, 6, 9, and 12) transplanted from the field into pots, which results in point clouds that are clear and free of extraneous objects. As the number of leaves is known, the instance segmentation is simplified to semantic segmentation with a fixed number of classes. They proposed using a PointNet structure for the segmentation of organs.

Luo et al. (2022) developed Eff-3DPSeg, a weakly supervised framework for segmenting plant organs in soybean to extract plant phenotypic traits at the organ level. Like DeepSeg3DMaize, they used a multiview stereo to produce point clouds from the crops grown in separate pots. Their model consists of two parts: a self-supervised SparseConvUnet model with a Viewpoint Bottleneck loss function (Tian et al., 2022; Luo et al., 2021) that learns plant representations from unlabeled point clouds and an instance segmentation head with convolution layers that are added to the SparseConvUnet backbone and fine-tuned to cluster the points into different plant organs. Several other similar works exist, including Label3DMaize (Miao et al., 2021), which uses Markov Random Fields to offer a semi-automatic toolkit for annotating point clouds obtained from transplanted maize using a multiview stereo technique; PlantPart (Shi et al., 2019), which introduces a multiview approach using 2D images for instance segmentation of isolated individual plant point clouds; and the work of Wang et al. (2022), which focuses on organ-level instance segmentation of plant point clouds using a structure similar to the one proposed in PartNet (Mo et al., 2019b).

The instance segmentation methods mentioned previously use simple structures that may lose important structural information between the points, leading to suboptimal results compared to more advanced methods like PlantNet (Li et al., 2022a) or TreePartNet (Liu et al., 2021b). PlantNet presents a multitasking model that simultaneously performs semantic and instance segmentation on a dataset of various plant species grown in separate pots. Their proposed neural network structure has two branches for semantic and instance segmentation tasks. It comprises a shared encoder that includes a series of EdgeConv layers, two decoder MLP-based pathways to up-sample encoded space, a feature fusion module that merges the instance and semantic segmentation embedding spaces, and a loss function similar to SGPN. Additionally, they propose a 3D Edge-Preserving Sampling (3DEPS) method for down-sampling the point cloud that preserves edge information. PlantNet's instance segmentation performance was evaluated using mean coverage, average precision, and average recall, which we have also used in our experiment section.

Liu et al. (2021b) proposed an innovative and advanced approach to instance segmentation of plant organs called TreePartNet. They designed two neural networks: one to semantically segment tree point clouds to remove the foliage and the other to segment various instances

of tree branches, which is the focus of this paper. The authors recognized that labeling a large number of 3D point clouds is impractical, so they trained their models on a synthetically generated dataset consisting of surface point samples from procedural tree models. To address the challenge of segmenting tree point clouds into an unknown number of branches, they utilized PointNet++ to learn per-point contextual features and create an initial fine clustering with a fixed number of clusters representing different parts of a branch. These initial clusters were merged using a jointly learned affinity matrix to form the final clusters. We provide a comprehensive comparison of our method to this approach.

1.3. Objectives and contributions

The main objective of this work is to develop a novel algorithm for point cloud instance segmentation that we call PlantSegNet. Our main contributions are as follows:

1. A pipeline for generating procedural sorghum models, which we then use to generate point clouds of sorghum fields.
2. A large-scale, annotated, and synthetic dataset of sorghum plants.
3. PlantSegNet model, a neural network structure for instance segmentation of nearby objects with identical semantics.
4. A demonstration of the effectiveness of training on synthetic data in learning an efficient feature embedding space for real data.

Additionally, our proposed technique addresses the challenge of instance segmentation of nearby objects with similar geometric structures. We provide details on our proposed method in Section 2, which includes the synthetic dataset generation pipeline, preprocessing models, and the PlantSegNet model. Section 3 summarizes our experiments, parameters, and results, and Section 5 concludes our paper.

2. Proposed method

To measure the structural characteristics of plants, we need to separate their organs from 3D point clouds. The motivation behind this research is segmenting sorghum leaves in 3D point clouds of plants grown densely in outdoor field settings, which presents several challenges. First, individual plants are difficult to isolate in the point clouds due to the dense planting, mutual overlaps, and shading in the lidar scanning. We address this by proposing a neural network structure as part of the preprocessing step to semantically segment the point cloud into three classes: focal plant, non-focal plant, and ground. This allows us to extract the focal plant and remove unwanted objects in the scene. Second, labeling the point clouds for deep learning training is a major obstacle due to the density of the canopies and their proximity to each other. It is also a very tedious and error-prone task if performed by humans. To overcome this, we create procedural developmental models of our field with the crops and use them to generate synthetic point clouds for annotation. We can also generate the corresponding labels since we have complete control over the generated virtual sorghum. The annotated point clouds are then used to train our proposed neural network models, allowing us to perform accurate organ segmentation at no additional cost.

2.1. Synthetic data generation

The realistic sorghum digital twin should include two components: plant geometry (shape and connectivity) and color. However, the point clouds in our approach only process the plant geometry, so we do not include the plant color.

2.1.1. Sorghum digital twin

We have developed a versatile synthetic 3D sorghum model that can procedurally generate unlimited 3D sorghum geometries with parts separated for labeling. Moreover, we also developed the gantry scanning simulator that generates point cloud data similar to those generated by the gantry.

Generating realistic shapes is a complex task, and we bootstrap our simulator by calibrated sorghum skeletons reconstructed by the algorithm of Gaillard et al. (2020) that is depicted in Fig. 2 (a) on the right. Each curve is then used as a skeletal curve of a generalized open cylinder (a sheet) corresponding to the sorghum leaves. The sheet geometry is generated as a spline surface with high triangular tessellation (Li et al., 2021, 2023; Lee et al., 2024). Each leaf is parameterized by adding the waviness and the length of the sheat and auricle. The leaves are assembled into the plant by positioning them as the characteristic co-centric pattern shown in Fig. 2 (b). The sorghum digital twin model has been calibrated by 10k sorghum plants reconstructed from real plants from a phenotyping facility (Gaillard et al., 2020). While the synthetic model will vary from real plant morphology, particularly in small details, the generated geometries are sufficient for the task presented in this work because they are used to train a deep neural model.

Once detailed sorghum plants are generated, we can create a virtual field. We randomly choose one of the geolocations of all sorghum in the field as the world origin, instantiate one procedural 3D sorghum at the origin, and also a dozen procedural sorghums with the same descriptor manually designed to fit the appearance and features of the real sorghum in the field. By placing the synthetic 3D sorghums in the field according to the list of measured geolocations of sorghum instances provided by the gantry system, we can provide identical spacing and arrangement of the sorghums for each synthetic point cloud. Over 5k sorghum geolocations are recorded for the field. By varying the sorghum rotation and adding variance to the descriptors, we generate an unlimited amount of synthetic point clouds with plant parts labeled (Fig. 3).

2.1.2. Point cloud generation

The synthetic point cloud is generated using an in-house developed ray tracer that imitates the features of the laser scanners of the gantry system. The gantry system has a stereo laser system with a resolution of at least 0.5 mm for all axes. The laser plane of both laser scanners has an angle of 33.5 ° to the normal direction of the ground surface (see Fig. 4).

Our virtual laser scanner provides similar functionalities by using the internal ray tracer to scan the target sorghum field from two directions to simulate the real point cloud with occlusions. The virtual scanner is set up according to the gantry specifications with an identical resolution to provide synthetic point cloud data close to the field measurement. The gantry system also provides accurate geolocation of each sorghum in the field with high-resolution latitude and longitude. For the dataset, 64k labeled point clouds were generated from 64k different 3D sorghum plants as the main plant and ≈1.2M different 3D sorghum plants as surrounding sorghum plants. The point cloud was downsampled to a resolution of 7.5 mm to capture necessary information while maintaining valid file sizes. Each point cloud is 5MB. The age of the procedurally generated plants was set to the age of the scanned real sorghum. The number of leaves for each plant ranges from 10–20.

2.2. Preprocessing and semantic segmentation model

To perform instance segmentation of leaves in point clouds, we first remove unwanted objects in the scene, such as ground and non-focal plants. We utilize a neural network that assigns a semantics identifier to each point in the point cloud. The network consists of sequential blocks of EdgeConv layers shown in Fig. 5. EdgeConv is an

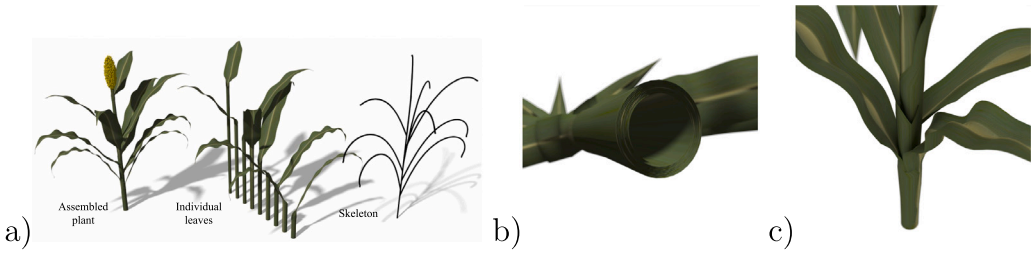

Fig. 2. The sorghum digital twin is a set of co-centric leaves (a) a detailed view in (b–c).

Fig. 3. Top view of the part of the virtual sorghum field captured within the framework (left), and a side view (right).

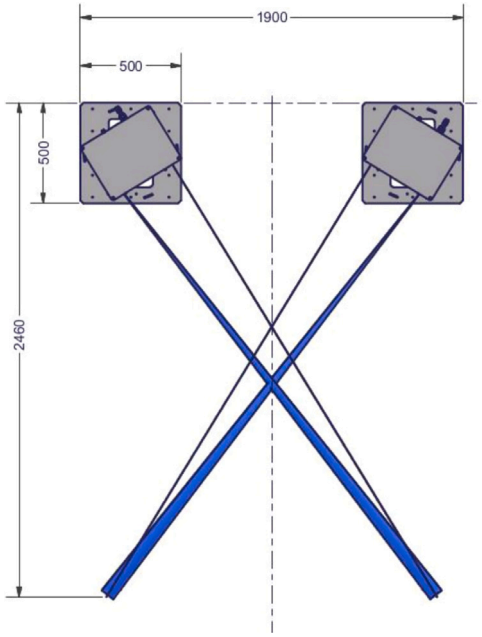

Fig. 4. The structure of the laser scanner used by the gantry system to provide high accuracy point clouds.

approximation of the convolution operation for point clouds, which is capable of capturing local structural features. This layer is used in all of our implemented neural network architectures. The EdgeConv operation dynamically finds the K nearest neighbors of each feature point in its input space and forms a graph with these neighbors as the nodes. It then forms edges between the main point and each neighbor and calculates edge features using an MLP. Finally, it applies max pooling on these edge features as a symmetry function to make the operation permutation invariant. The computation in an EdgeConv layer is summarized in the following equations:

$$f_{ij}^{l} = LReLU\left(\theta_l(x_i^{l-1} - x_j^{l-1} \oplus x_i^{l-1})\right)$$
$$x_i^{l} = max_{j\in N(i)}\{f_{ij}^{l}\}, \quad (1)$$

where x_i^l is the ith point in the output space of the lth layer, and the operator $\oplus$ concatenates vectors. The parameters of the MLP used to learn features are represented by θ, and f_{ij}^l is the edge feature between points i and j. The activation function $LReLU$ is used, and $N(i)$ is the list of K nearest neighbors of point i. The vectors concatenated by $\oplus$ convey local and global structural information to the network, enabling the network to learn more effective embedding features for each point.

2.3. PlantSegNet model

Instance segmentation is a challenging task due to the unknown number of instances in a scene. Different methods have addressed this issue in various ways. For example, TreePartNet (Liu et al., 2021b) assumes a fixed upper limit on the possible number of instances, segments the point cloud into that number of clusters, and then merges the clusters based on their similarity. Other methods like SGPN (Wang

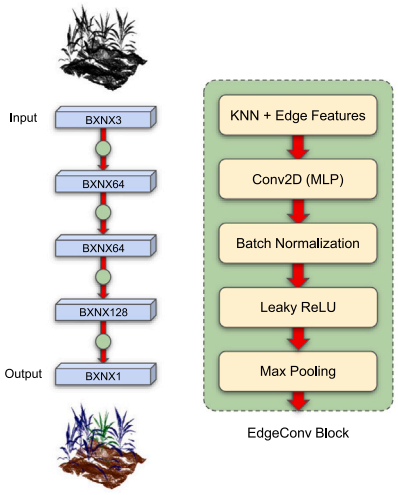

Fig. 5. The structure of the semantic segmentation model used in preprocessing the data for removing the points associated with the ground and non-focal plants.

et al., 2018) reduce instance segmentation to the problem of learning a feature embedding space, where points associated with the same instance are close to each other. We introduce PlantSegNet, a neural network for point-based instance segmentation with a novel loss function that adopts the latter approach to address the challenges of this task.

In contrast to SGPN (Wang et al., 2018), which uses a similar approach to our method to address the unknown number of instances, we chose not to combine semantic segmentation with instance segmentation in a single network since we found that the required features are distinct between the two tasks in our specific scenario. Furthermore, instead of using PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), as in SGPN, we utilized EdgeConv operations as the building blocks of our instance segmentation network, as we found them to be highly effective in learning structural features. Using a sequence of EdgeConv operations and our novel, robust loss function, PlantSegNet learns a latent space in which the points belonging to the same instance are situated close together. Ultimately, a similarity matrix is calculated in this space between all point pairs, which is leveraged during the training phase for estimating the loss function. Fig. 6 illustrates the proposed structure for instance segmentation.

2.3.1. Robust loss function

We propose a robust loss function to address the challenge of ambiguity in the feature space for points that belong to different instances but are in close proximity. This is a hurdle in instance segmentation by learning embedding spaces. Points on instance boundaries may be difficult to separate in the embedding space, especially around junction points. To tackle this issue, our proposed loss function penalizes mistakes more strictly in the areas around the junction points and on the points close to each other, but not in the same instance. We accomplish this by defining a piecewise function:

$$L_{ij} = \begin{cases} |F_i - F_j|_2, & C_i = C_j \\ \frac{|P_i - P_j|_2^{-1}}{|F_i - F_j|_2}, & C_i \neq C_j \end{cases}$$

$$L = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} L_{ij}, \tag{2}$$

where F_i represents the learned feature embedding for a given point i, C_i is the ground truth cluster label for that point, and P_i is the coordinate of the point in 3D space, L_{ij} represents the loss between the pair of points i and j, N is the total number of points, and L is the total loss which is the average over all possible pairs. By minimizing the total loss value L, we optimize for the best parameters of the network.

This function applies a penalty to the model that is proportional to the distances between the feature vectors of points belonging to the same instance and inversely proportional to the feature vector distance for points belonging to different clusters. To impose more severe penalties on the poor feature vectors of points in the junction areas, the model adds more penalties by multiplying the inverse of the Euclidean distance between the point locations.

In contrast to the double-hinge loss function proposed in SGPN, which does not penalize the feature vector difference between points that fall below or above a threshold, our continuous loss function pushes the pairwise distances to their appropriate limits. It applies even stricter penalties on the points around the junction areas. Moreover, points in the hinge loss function domain with undefined left and right derivatives may negatively impact the training optimization.

2.4. Final clustering during inference and test

The PlantSegNet model transforms points into a feature embedding space where instances can be more easily distinguished. However, the model does not directly provide a cluster label for each point; additional processing of the embedding space is required for this purpose.

Various clustering methods can be applied to assign per-point instance labels. We use DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996) to create cluster labels from the embedding space. DBSCAN groups the points sufficiently close to one another and have a minimum number of points in the cluster. This step is only performed during inference to generate prediction labels and evaluate performance metrics.

3. Experiments and results

To test the efficacy of our proposed model, we designed a series of experiments and measured the performance of our proposed method on two different datasets. We compared it to the performance of TreePartNet, the method closest to our use case in terms of application in the literature. All the experiments are run on a machine with an AMD EPYC 7542 32-core processor, 1008 GB of RAM, and two NVIDIA A100 GPUs, each with 40 GB of dedicated memory. The code for our model, as well as the wrapper for TreePartNet (Liu et al., 2021b), were developed using PyTorch, and it is publicly available at https://github.com/ariyanzri/PlantSegNet alongside our labeled synthetic and real data.

3.1. Dataset

Our primary dataset, which we refer to as the Synthetic Sorghum Dataset, comprises 64,000 synthetically generated point clouds of sorghum plants. We split this dataset into training, validation, and test sets with sizes 38,400, 12,800, and 12,800, respectively. The point clouds come with two sets of labels, i.e., the per-point semantic annotations used to segregate the focal plant from non-focal plants and ground points and the per-point instance annotations that determine the leaf instance to which each point belongs. All the point clouds are normalized before being used in the models using the min–max normalization method separately on each dimension of the 3D space. Also, all the point clouds are uniformly downsampled to have 8,000 points, so we can fit them in batches and meet GPU memory limitations. Additionally, to check the performance of the models on unseen data that represent real sorghum plants, we prepared another small dataset called the Real Sorghum Dataset. We manually labeled seven point clouds in this dataset, which are captured and generated by a laser scanner from sorghum plants grown in an outdoor field setting.

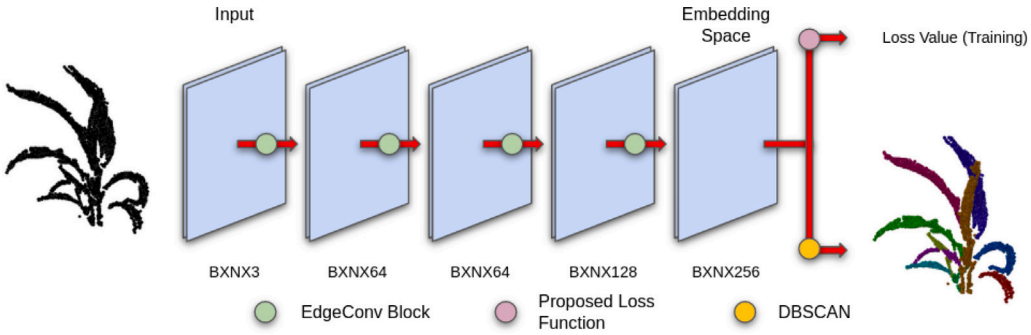

Fig. 6. The structure of the PlantSegNet model. The EdgeConv block used in the PlantSegNet model is identical to the one used in the semantic segmentation model shown in Fig. 5. We utilize a novel loss function described in Section 2.3.1 to calculate the loss value and optimize the model weights during the training phase. We apply the DBSCAN algorithm on the embedding space to assign instance labels to the points during the inference phase.

Furthermore, we conducted a comparative analysis between our proposed method and TreePartNet (Liu et al., 2021b) using our sorghum data. To make this possible, we created a modified version of the synthetic and real sorghum datasets, following the conventions of TreePartNet. Moreover, we wanted to demonstrate that our model is versatile and capable of working well with other 3D objects. To that end, according to PlantSegNet input format requirements, we developed a modified version of the TreePartNet paper's dataset named the Tree Dataset. The Tree Dataset includes 3,521, 440, and 440 point clouds in the training, validation, and test sets. These datasets are now publicly available on our GitHub page https://github.com/ariyanzri/PlantSegNet.

3.2. Data augmentation

To enhance the diversity of our artificially generated dataset and make it more resilient to the noises present in the real data, we introduce a common noise to each point coordinate in all three dimensions of the 3D space separately. The noise has a mean of zero and a standard deviation of 0.01. Furthermore, to improve the effectiveness of the semantic segmentation task, we randomly remove non-focal plants from 20% of the point clouds during the training process, which helps to generate additional point clouds containing only a single plant.

3.3. Metrics

We evaluated the performance of our models using commonly used metrics for point cloud instance segmentation. In particular, we used Average Precision (AP), Average Recall (AR), and Mean Coverage (mCov) (Li et al., 2022a). Moreover, we proposed a new set of pointwise metrics to provide a different perspective on instance segmentation. Specifically, we introduced Pointwise Precision (PWP), Pointwise Recall (PWR), Pointwise Accuracy (PWA), and Pointwise F1 Score (PWF) to evaluate the models' effectiveness at the point level. By using both traditional and pointwise metrics, we gained a comprehensive understanding of our models' performance.

To compute the instance-level metrics, we must first calculate the Intersection over Union (IoU) between the predicted and ground truth instances. Precision and Recall at the instance level are typically reported at specific IoU thresholds, such as 0.5 or 0.75, as in Li et al. (2022a). The following formulas are used to calculate these metrics:

$$Pr_{th} = \frac{TP}{PI} \quad Re_{th} = \frac{TP}{GI}, \tag{3}$$

Pr_{th} and Re_{th} indicate respectively the instance-level precision and recall with IoU threshold th. The total number of predicted and ground truth instance pairs with an IoU above a specified threshold (th) is represented by TP. In contrast, PI represents the total number of predicted instances and GI represents the total number of ground truth instances. Average Precision, AP, and Average Recall, AR, are then computed by averaging instance-level Precision and Recall across various IoU thresholds:

$$AP = \frac{1}{T}\sum_{th \in T} Pr_{th} \quad AR = \frac{1}{T}\sum_{th \in T} Re_{th}, \tag{4}$$

where the list of all IoU threshold values is represented by T. In our experiments, we calculated AP and AR using IoU thresholds of 0.25, 0.5, and 0.75. Mean Coverage $mCov$, on the other hand, is calculated as:

$$mCov = \frac{1}{GI}\sum_{i=1}^{GI} max_{j=1}^{|PI|} IoU(GI_i, PI_j), \tag{5}$$

where GI_i and PI_j refer to the ith ground truth and jth predicted instances, respectively. While AP and AR have been widely used for instance segmentation and object detection, we devised a different approach to evaluate the performance of instance segmentation models at the individual point level by expanding the definitions of TP, FP, TN, and FN to all pairs of points. The four values for each point pair are calculated as:

$$\begin{aligned} TP &= \left|\left\{(i,j), \forall (i,j) \in [1,N)^2 \,|C_i = C_j \wedge O_i = O_j\right\}\right| \\ FN &= \left|\left\{(i,j), \forall (i,j) \in [1,N)^2 \,|C_i = C_j \wedge O_i \neq O_j\right\}\right| \\ FP &= \left|\left\{(i,j), \forall (i,j) \in [1,N)^2 \,|C_i \neq C_j \wedge O_i = O_j\right\}\right| \\ TN &= \left|\left\{(i,j), \forall (i,j) \in [1,N)^2 \,|C_i \neq C_j \wedge O_i \neq O_j\right\}\right|. \end{aligned} \tag{6}$$

In this context, a "positive" refers to two points that belong to the same instance. Using the expanded definitions of TP, FN, FP, and TN, we compute pointwise Accuracy, Precision, Recall, and F1 score according to their original definitions. While these new metrics offer an alternative perspective on the results of instance segmentation, it is important to note that the outcomes of AP and AR are still more strongly correlated with qualitative assessments of instance segmentation.

3.4. Semantic segmentation

One primary preprocessing step is extracting the focal plant from the point clouds, which involves semantically segmenting the point clouds into classes such as focal plant, non-focal plants, and ground. We trained the semantic segmentation model described in Section 2.2 on the synthetic dataset. The training process is carried out for 50 epochs with a batch size of four, a learning rate of 1×10^{-4}, and an Adam optimizer. We set $K = 15$ in the EdgeConv KNN module and applied early stopping on validation loss with a patience value of 20. The Categorical Cross Entropy loss function is used for this network. To evaluate the performance of our semantic segmentation model, we conducted both quantitative and qualitative assessments on the synthetic and real test datasets. The accuracy achieved by our model on

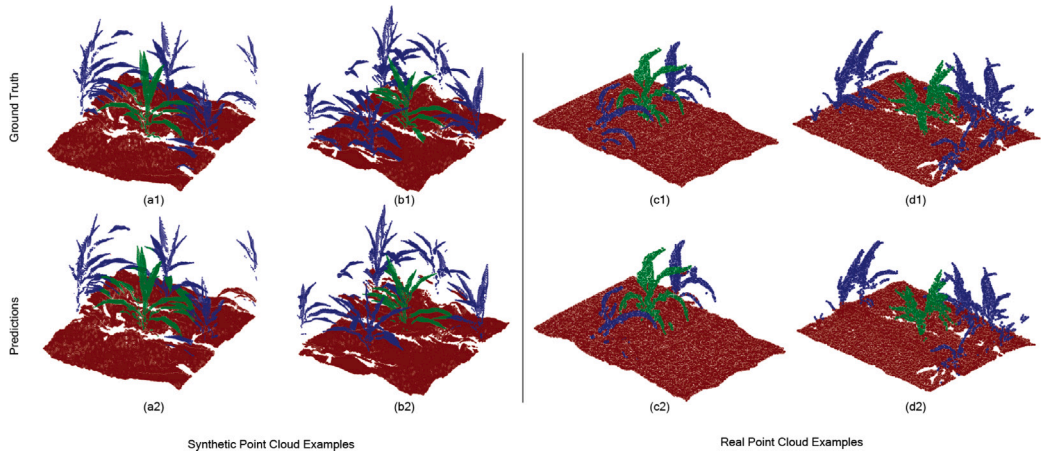

Fig. 7. Qualitative results of semantic segmentation model on instances of the synthetic (left) and real (right) data. The top row shows the point clouds, colored with the ground truth annotations, and the bottom row represents the predictions of the model. Columns (a) and (b) correspond to the synthetic test set, and columns (c) and (d) to the real data.

Table 1
The performance of the PlantSegNet model on real and synthetic sorghum data with two loss functions: the proposed function and the hinge loss function. Our proposed loss function is designed to penalize predictions of nearby point pairs that belong to different instances more strictly. This function outperformed the hinge loss function across all the metrics on the real data and in terms of AP and PWR on the synthetic test data.

Dataset	Loss function	mCov	AP	AR	PWA	PWP	PWR	PWF
Synthetic Test Set	Proposed	0.68	**0.72**	0.70	0.95	0.71	**0.92**	0.79
	Hinge (SGPN)	**0.75**	0.68	**0.79**	**0.97**	**0.83**	0.89	**0.85**
Real Data	Proposed	**0.53**	**0.69**	**0.55**	**0.89**	**0.50**	**0.78**	**0.60**
	Hinge (SGPN)	0.51	0.56	0.53	0.89	0.47	0.72	0.55

the test set of the synthetic and real datasets were 99.35% and 95.27%, respectively. Fig. 7 displays the qualitative results of our model on two samples of each dataset (synthetic and real), which demonstrates the performance of our semantic segmentation model.

3.5. Instance segmentation

This section presents the outcomes of our proposed PlantSegNet model for instance segmentation, tested on different datasets and in various experiments. We exclusively trained the model on focal plants during all the instance segmentation trials on sorghum. Using the ground truth labels, we eliminated non-focal plants and ground from the training data.

3.5.1. Loss function experiments

To evaluate the effectiveness of our suggested loss function and contrast it with the hinge loss function employed in SGPN, we conduct training on two variations of our PlantSegNet architecture on our synthetic sorghum dataset. The first model incorporates the proposed loss function discussed in Section 2.3.1 while the other uses a hinge loss function, defined by:

$$
\begin{aligned}
D_{ij} &= |F_i - F_j|_2 \\
L_{ij} &= \left\{ \begin{array}{ll} max(0, D_{ij} - M1), & C_i = C_j \\ max(0, M2 - D_{ij}), & C_i \neq C_j \end{array} \right\} \\
L &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} L_{ij}.
\end{aligned} \tag{7}
$$

This particular loss function ensures that all points within a cluster have a distance below $M1$ in the embedding space, while points belonging to different instances have a distance above $M2$. This approach is highly reminiscent of the loss function used in the SGPN paper.

We train both versions of PlantSegNet, using the proposed loss function and the hinge loss function, for 50 epochs. The training involves using an Adam optimizer with a learning rate of 1×10^{-4}, a batch size of 4, and a K value 25 in the EdgeConv KNN module. We empirically set $M1$ and $M2$ to 1 and 10, respectively. We perform early stopping on validation loss to prevent overfitting, with a patience of 20.

Furthermore, we fine-tune the hyperparameters of both models and select the optimal DBSCAN parameters based on the average of AP and AR scores on the validation set. The PlantSegNet's instance segmentation performance using the two loss functions is summarized in Table 1 for both synthetic test data and real data. It is evident from the results that our proposed loss function surpasses the hinge loss function from the SGPN paper across all metrics on real data and for AR and PWR on the synthetic test set. This highlights the effectiveness of our proposed loss function compared to the hinge loss function.

To provide a more thorough evaluation of the model's performance using different loss functions, Fig. 8 presents the results for two instances from the synthetic test data and two instances from the real data, including the models' output, the 2-component PCA of the embedding space, and the ground truth labels. The 2-component PCA visualizations reveal that our proposed loss function produces a more distinct embedding space, making it easier to separate the data points that belong to different instances. Considering these findings and the better performance shown in the quantitative results in Table 1, we have decided to utilize this particular loss function in all the upcoming experiments.

3.5.2. Experiments on our dataset

In this section, we evaluate the performance of our proposed model and compare it to that of TreePartNet using both synthetic and real sorghum datasets that we have included with this paper. To run experiments with TreePartNet, we use the code from Liu et al. (2021a). We trained TreePartNet for 100 epochs with a learning rate of $5{\times}10^{-2}$ and a batch size of 2. For the other parameters, we followed the suggestions in the TreePartNet paper. Similarly, we trained PlantSegNet for 50 epochs with a learning rate of 1×10^{-4}, a batch size of 4, and using the Adam optimizer. We also set the value of K in the EdgeConv KNN module to 25. Additionally, we utilized early stopping on validation loss with the

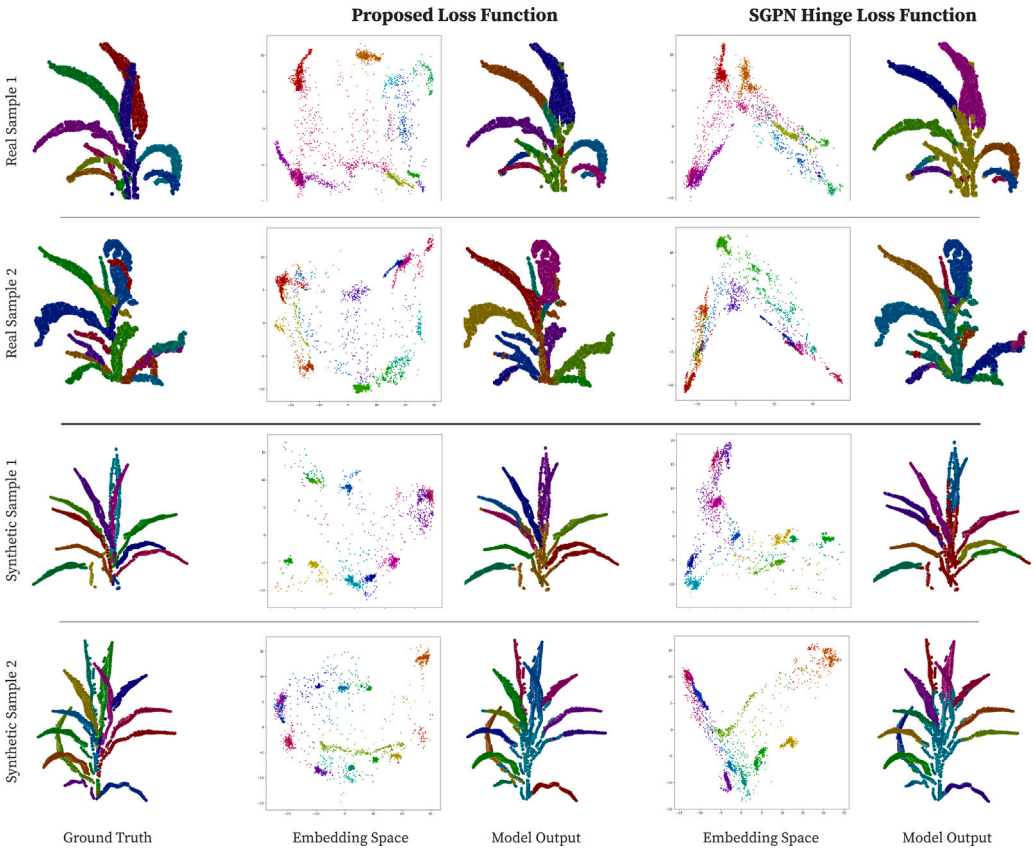

Fig. 8. A comparison between the proposed loss function and the SGPN hinge loss function on real and synthetic data samples. The columns, from left to right, represent (1) the ground truth instances, 2 and (3) 2-component PCA visualizations of the embedding space of the model with the proposed loss function on the given point cloud and its respective model output, and 4 and (5) 2-component PCA visualizations of the embedding space of the model with the SGPN hinge loss function on the given point cloud, and its respective model output. The proposed loss function results in a more separable embedding space than the hinge loss function. Additionally, the proposed loss function performs better on points near the stem, a junction area, whereas the hinge loss function merely clusters them together. Note that colors in the PCA images correspond to the ground truth annotations.

Table 2
The table compares the performances of PlantSegNet and TreePartNet on real and synthetic sorghum data. PlantSegNet outperforms TreePartNet on all metrics for both real and synthetic data. The significant difference in AP and AR between the two models demonstrates how well PlantSegNet's predicted instances match the ground truth.

Dataset	Model	mCov	AP	AR	PWA	PWP	PWR	PWF
Synthetic Test Set	PlantSegNet	**0.68**	**0.72**	**0.70**	**0.95**	**0.71**	**0.92**	**0.79**
	TreePartNet	0.30	0.38	0.24	0.79	0.29	0.72	0.41
Real Data	PlantSegNet	**0.53**	**0.69**	**0.55**	**0.89**	**0.50**	**0.78**	**0.60**
	TreePartNet	0.17	0.16	0.09	0.65	0.18	0.61	0.27

patience of 20 and applied the data augmentation technique described in Section 3.2 to both models.

After training PlantSegNet on the synthetic sorghum dataset, we fine-tuned the DBSCAN parameters by maximizing the average of AP and AR on the validation set. Our results for both PlantSegNet and TreePartNet on the synthetic test set and the real data using instance-based and point-based metrics are summarized in Table reftable:ins-sorghum-res. PlantSegNet outperforms TreePartNet on all metrics for both real and synthetic data. Fig. 9 visually compares the two models and supports the findings from Table 2.

3.5.3. Experiments on TreePartNet dataset

To demonstrate the versatility and generalizability of our proposed model in processing various 3D objects regardless of their structures, we conducted an evaluation of its performance on tree point clouds from the dataset published by the creators of TreePartNet. We then compared its results against TreePartNet accessible at Liu et al. (2021a).

We trained PlantSegNet on the training set for 100 epochs with a batch size of 4, a learning rate of 1×10^{-4}, and a DGCNN kernel size 25. Similarly, we trained TreePartNet for 100 epochs using a batch size of 2 and a learning rate of 5×10^{-2}. We also employed early stopping on the validation loss, with a patience of 20 epochs. After tuning the DBSCAN parameters over the validation set for PlantSegNet, we evaluated the performance of these models on the test set.

Table 3 presents the numerical outcomes of the two models' performance on the tree dataset. It appears that PlantSegNet outperforms TreePartNet by a considerable margin across all metrics. Fig. 10 depicts a qualitative evaluation of the effectiveness of PlantSegNet and TreePartNet in segmenting instances of tree branches. The visual evidence shows that PlantSegNet produces more precise instances that closely match the ground truth instances.

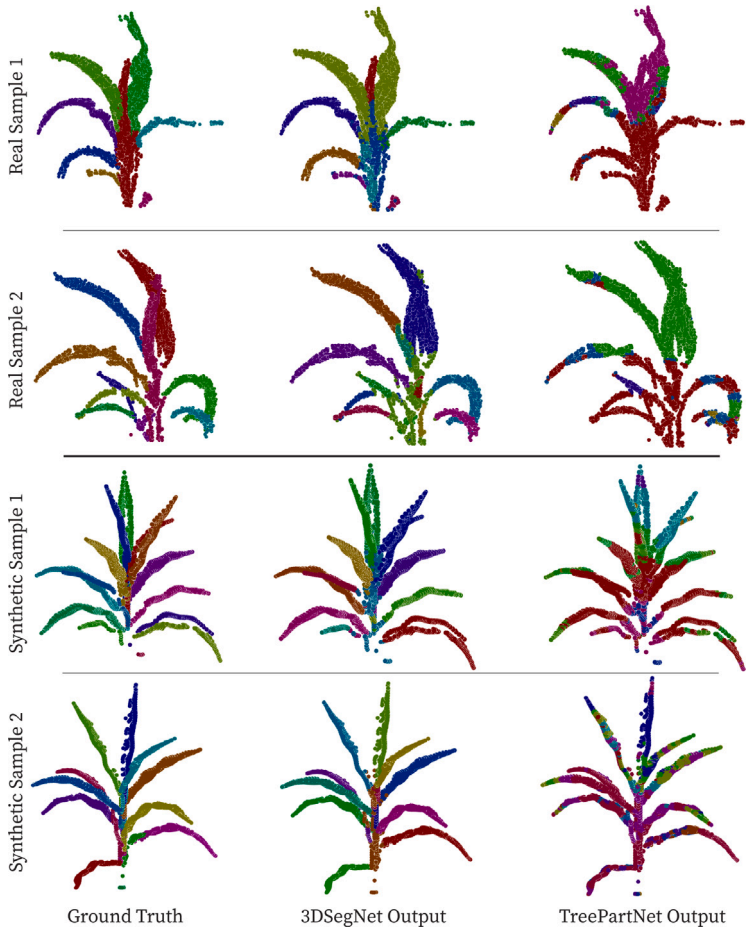

Fig. 9. A visual assessment of the performances of PlantSegNet and TreePartNet on both the synthetic and real sorghum datasets. It presents the ground truth and predicted instance labels on two samples from the real and two samples of the synthetic datasets. Our findings suggest that our proposed model generates more precise predictions for the instance labels of the points compared to TreePartNet, particularly in areas where nearby points belong to different instances, like the regions close to the stem.

Table 3
The quantitative outcomes of testing PlantSegNet and TreePartNet on the tree dataset. PlantSegNet is 1.5-3 times better than TreePartNet in all the metrics.

	PlantSegNet	TreePartNet
mCov	**0.51**	0.13
AP	**0.61**	0.17
AR	**0.55**	0.06
PWA	**0.86**	0.82
PWP	**0.40**	0.28
PWR	**0.90**	0.71
PWF	**0.53**	0.39

4. Discussion

We introduced PlantSegNet to address the challenges of segmenting individual leaves in point clouds of sorghum plants grown in outdoor field settings and also to provide a novel approach for the general problem of instance segmentation of nearby objects with similar geometric structures. The ultimate objective of the underlying research problem is to use the segmented point clouds to analyze and measure the phenotypic information of these plants as they grow in an outdoor field under stressed conditions. Our proposed method involves (1) a pipeline for generating procedural sorghum models, (2) a large-scale, annotated synthetic point cloud dataset of sorghum plants generated using the synthetic sorghum models, and (3) a neural network structure for instance segmentation of nearby objects with identical semantics.

To assess the effectiveness of our proposed model, we conducted multiple experiments using both sorghum and tree point cloud data. In our comparisons, we used TreePartNet as it closely relates to our application use case. Furthermore, we evaluated the performance of our proposed loss function by comparing it against the loss function introduced in the SGPN method.

Our results, as shown in Table 1, demonstrate that our proposed loss function achieves higher scores in all metrics on the unseen real dataset. Additionally, we visualized the learned feature embeddings using 2-component PCA (Fig. 8) and found that our proposed loss function produces a much better embedding space. In this space, points belonging to different instances are well-separated, in contrast to the hinge loss proposed in SGPN.

Finally, we compared PlantSegNet against TreePartNet on the sorghum and tree datasets. The comparison indicated that PlantSegNet outperforms TreePartNet on both datasets. Specifically, PlantSegNet achieved an average precision of 0.69 on the sorghum real dataset, while TreePartNet only achieved 0.16. Similarly, PlantSegNet achieved an average precision of 0.61 on the tree dataset, about four times higher than the average precision of 0.17 achieved by TreePartNet. The qualitative results shown in Figs. 9 and 10 further confirm these findings.

Limitations: Our proposed algorithm has several limitations. First, it depends on the synthetic digital twins that were trained on a particular Sorghum, and it could be less precise if applied to plants that are not captured by the digital twin. The second limitation comes from the precision of the data acquisition system, as there are physical limits to what can be captured. The third limitation is given by the neural model. While we used the current state-of-the-art algorithms, they may

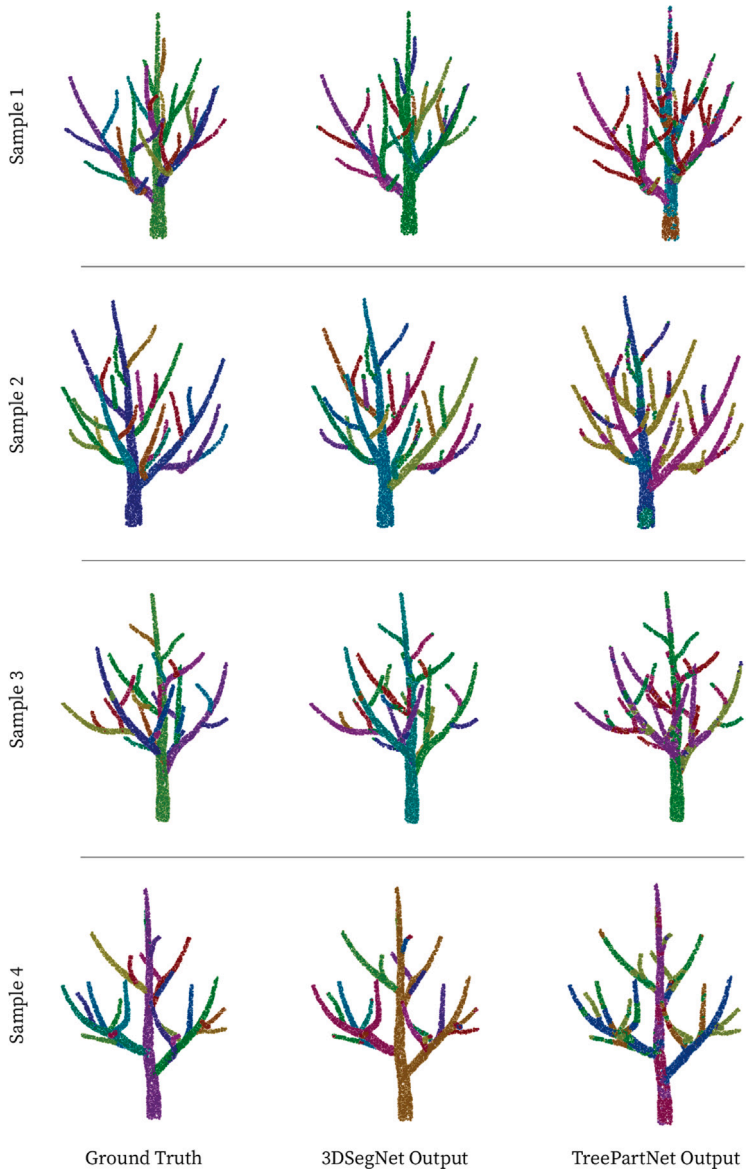

Fig. 10. Comparison of PlantSegNet and TreePartNet in segmenting instances of branches in the tree dataset proposed by the TreePartNet paper. A detailed examination of the predicted instance labels reveals that PlantSegNet generates more accurate instances, while TreePartNet sometimes generates multiple instances for a single branch.

not capture all details and may have some intrinsic limitations that could affect the results, such as misclassification.

Future work: There are many possible avenues for future work. One promising extension is in using a different baseline deep neural model for point classification. For example, a concurrent work (Bai et al., 2024) introduces the segmentation of wood and leaves from point clouds. Using some of the transformed-based point classifiers, such as the Point Transformer (Zhao et al., 2021), would also be interesting.

5. Conclusions

We proposed a novel approach for segmenting individual leaves in point clouds of sorghum plants grown in outdoor field settings. This can be generalized to the problem of instance segmentation of nearby stochastic geometries with a similar structure. Our proposed method uses a digital twin synthetic data generation technique and deep learning-based instance segmentation. Our newly introduced PlantSegNet allows us to address the challenge of acquiring labeled data, which is time-consuming and labor-intensive for 3D point clouds, by generating a large synthetic dataset using realistic models of plants. The performance of our proposed method was evaluated on both synthetic and real datasets, and it was shown to outperform the state-of-the-art method, TreePartNet, in terms of segmentation accuracy.

Our method can potentially improve the field of plant phenotyping by enabling high-throughput, automated analysis of plant structures and characteristics, especially in outdoor field settings. It also opens up opportunities for various applications, such as precision agriculture and crop management, by accurately and efficiently measuring plant phenotypes. In addition, the proposed synthetic dataset generation method can be helpful in other fields where labeled 3D point cloud data is limited. Overall, this work shows the potential of combining data synthesis and deep learning to improve the efficiency and accuracy of 3D point cloud analysis, especially in plant phenotyping.

CRediT authorship contribution statement

Ariyan Zarei: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Bosheng Li:** Data curation, Methodology, Software, Visualization, Writing – original draft, Writing – review

& editing. **James C. Schnable:** Funding acquisition, Writing – review & editing, Formal analysis. **Eric Lyons:** Conceptualization, Formal analysis, Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing. **Duke Pauli:** Conceptualization, Formal analysis, Funding acquisition, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing. **Kobus Barnard:** Supervision, Writing – original draft, Investigation, Project administration, Resources, Writing – review & editing, Conceptualization, Formal analysis, Funding acquisition. **Bedrich Benes:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

Authors state no conflict of interest.

Data availability

Link to the github page with code and data is available in the manuscript.

Acknowledgments

This research was supported by the Foundation for Food and Agriculture Research, United States Grant ID: 602757 to Benes and Schnable. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the Foundation for Food and Agriculture Research. This work was also supported by the Agricultural Genome to Phenome Initiative grant "Optimizing 3D Canopy Architecture for Better Crops" to Benes and Pauli. This work was supported by NRCS, United States grant #NR233A750004G044 to Benes. The findings and conclusions should not be construed to represent any agency determination or policy. This work is based upon efforts supported by the EFFICACI grant, #NR233A750004G044, under USDA NRCS to Benes. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government or NRCS. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein. This work was supported by the Department of Energy Advanced Research Projects Agency-Energy, award number DE-AR0000594 Bill and Melinda Gates Foundation, United States to Pauli, award number OPP1129603 Department of Energy Advanced Research Projects Agency-Energy, DE-AR0001101 to Pauli, Department of Energy Biological and Environmental Research, award numbers DE-SC0020401, DE-SC0023305 Cotton Incorporated, United States, award numbers 20-720, 21-830 to Pauli.

References

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1534–1543.

Bai, Y., Durand, J.B., Vincent, G., Forbes, F., 2024. Semantic segmentation of sparse irregular point clouds for leaf/wood discrimination. Adv. Neural Inf. Process. Syst. 36.

Duan, T., Chapman, S., Holland, E., Rebetzke, G., Guo, Y., Zheng, B., 2016. Dynamic quantifcation of canopy structure to characterize early plant vigour in wheat genotypes. J. Exp. Bot. 67 (15), 4523–4534.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, Vol. 96, No. 34. pp. 226–231.

Gaillard, M., Miao, C., Schnable, J., Benes, B., 2020. Sorghum segmentation by skeleton extraction. In: Bartoli, A., Fusiello, A. (Eds.), Computer Vision – ECCV 2020 Workshops. Springer International Publishing, Cham, pp. 296–311. http://dx.doi.org/10.1007/978-3-030-65414-6_21.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.

Hou, J., Dai, A., Nießner, M., 2019. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4421–4430.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8500–8509.

Lee, J.J., Li, B., Benes, B., 2024. Latent L-systems: Transformer-based tree generator. ACM Trans. Graph. 43 (1), http://dx.doi.org/10.1145/3627101.

Li, B., Kałużny, J., Klein, J., Michels, D.L., Pałubicki, W., Benes, B., Pirk, S., 2021. Learning to reconstruct botanical trees from single images. ACM Trans. Graph. 40 (6), 1–15. http://dx.doi.org/10.1145/3478513.3480525.

Li, B., Klein, J., Michels, D.L., Pirk, S., Benes, B., Palubicki, W., 2023. Rhizomorph: The coordinated function of shoots and roots. ACM Trans. Graph. 42 (4), http://dx.doi.org/10.1145/3592145.

Li, D., Shi, G., Li, J., Chen, Y., Zhang, S., Xiang, S., Jin, S., 2022a. PlantNet: A dual-function point cloud segmentation network for multiple plant species. ISPRS J. Photogramm. Remote Sens. 184, 243–263.

Li, Y., Wen, W., Miao, T., Wu, S., Yu, Z., Wang, X., Guo, X., Zhao, C., 2022b. Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning. Comput. Electron. Agric. 193, 106702.

Liang, Z., Yang, M., Li, H., Wang, C., 2020. 3D instance embedding learning with a structure-aware loss function for point cloud segmentation. IEEE Robot. Autom. Lett. 5 (3), 4915–4922.

Liu, Y., Guo, J., Benes, B., Deussen, O., Zhang, X., Huang, H., 2021a. TreePartNet github page. URL: https://github.com/marktube/TreePartNet.git.

Liu, Y., Guo, J., Benes, B., Deussen, O., Zhang, X., Huang, H., 2021b. TreePartNet: neural decomposition of point clouds for 3D tree reconstruction. ACM Trans. Graph. 40 (6).

Luo, L., Jiang, X., Yang, Y., Samy, E.R.A., Lefsrud, M., Hoyos-Villegas, V., Sun, S., 2022. Eff-3DPSeg: 3D organ-level plant shoot segmentation using annotation-efficient point clouds. arXiv preprint arXiv:2212.10263.

Luo, L., Tian, B., Zhao, H., Zhou, G., 2021. Pointly-supervised 3d scene parsing with viewpoint bottleneck. arXiv preprint arXiv:2109.08553.

Maturana, D., Scherer, S., 2015. VoxNet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 922–928.

Miao, T., Wen, W., Li, Y., Wu, S., Zhu, C., Guo, X., 2021. Label3DMaize: toolkit for 3D point cloud data annotation of maize shoots. GigaScience 10 (5), giab031.

Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H., 2019a. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition. CVPR.

Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H., 2019b. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 909–918.

Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y., Yuan, L., 2022. Masked autoencoders for point cloud self-supervised learning. In: European Conference on Computer Vision. Springer, pp. 604–621.

Pham, Q.H., Nguyen, T., Hua, B.S., Roig, G., Yeung, S.K., 2019. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8827–8836.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Riegler, G., Osman Ulusoy, A., Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3577–3586.

Shi, W., van de Zedde, R., Jiang, H., Kootstra, G., 2019. Plant-part segmentation using deep learning and multi-view vision. Biosyst. Eng. 187, 81–95.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 945–953.

Tian, B., Luo, L., Zhao, H., Zhou, G., 2022. VIBUS: Data-efficient 3D scene parsing with VIewpoint bottleneck and uncertainty-spectrum modeling. ISPRS J. Photogramm. Remote Sens. 194, 302–318.

Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X., 2017. O-CNN: Octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. 36 (4), 1–11.

Wang, X., Liu, S., Shen, X., Shen, C., Jia, J., 2019a. Associatively segmenting instances and semantics in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4096–4105.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. Acm Trans. Graph. (tog) 38 (5), 1–12.

Wang, W., Yu, R., Huang, Q., Neumann, U., 2018. SGPN: Similarity group proposal network for 3d point cloud instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2569–2578.

Wang, L., Zheng, L., Wang, M., 2022. 3D point cloud instance segmentation of lettuce based on PartNet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1647–1655.

Wei, X., Yu, R., Sun, J., 2020. View-GCN: View-based graph convolutional network for 3d shape analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1850–1859.

Wu, J., Cawse-Nicholson, K., van Aardt, J., 2013. 3D tree reconstruction from simulated small footprint waveform lidar. Photogramm. Eng. Remote Sens. 79 (12), 1147–1157.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1912–1920.

Wu, S., Wen, W., Wang, Y., Fan, J., Wang, C., Gou, W., Guo, X., 2020. MVS-Pheno: a portable and low-cost phenotyping platform for maize shoots using multiview stereo 3D reconstruction. Plant Phenomics 2020.

Xu, Q., Sun, X., Wu, C.Y., Wang, P., Neumann, U., 2020. Grid-GCN for fast and scalable point cloud learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5661–5670.

Yamada, R., Kataoka, H., Chiba, N., Domae, Y., Ogata, T., 2022. Point cloud pre-training with natural 3d structures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21283–21293.

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N., 2019. Learning object bounding boxes for 3D instance segmentation on point clouds. Adv. Neural Inf. Process. Syst. 32.

Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J., 2019. GSPN: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3947–3956.

Yu, T., Meng, J., Yuan, J., 2018. Multi-view harmonized bilinear network for 3d object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 186–194.

Zarei, A., Gonzalez, E., Merchant, N., Pauli, D., Lyons, E., Barnard, K., 2022. MegaStitch: Robust large-scale image stitching. IEEE Trans. Geosci. Remote Sens. 60, 1–9.

Zhang, Q., Hou, J., 2023. PointVST: Self-supervised pre-training for 3d point clouds via view-specific point-to-image translation. IEEE Trans. Vis. Comput. Graphics.

Zhang, Q., Hou, J., Qian, Y., 2023a. PointMCD: Boosting deep point cloud encoders via multi-view cross-modal distillation for 3d shape recognition. IEEE Trans. Multimed..

Zhang, Q., Hou, J., Qian, Y., Chan, A.B., Zhang, J., He, Y., 2022. RegGeoNet: Learning regular representations for large-scale 3d point clouds. Int. J. Comput. Vis. 130 (12), 3100–3122.

Zhang, Q., Hou, J., Qian, Y., Zeng, Y., Zhang, J., He, Y., 2023b. Flattening-net: Deep regular 2d representation for 3d point cloud analysis. IEEE Trans. Pattern Anal. Mach. Intell..

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268.

Zhou, H., Feng, Y., Fang, M., Wei, M., Qin, J., Lu, T., 2021. Adaptive graph convolution for point cloud analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4965–4974.