



HAL
open science

RoseSegNet: An attention-based deep learning architecture for organ segmentation of plants

Kaya Turgut, Helin Dutagaci, David Rousseau

► To cite this version:

Kaya Turgut, Helin Dutagaci, David Rousseau. RoseSegNet: An attention-based deep learning architecture for organ segmentation of plants. *Biosystems Engineering*, 2022, 221, pp.138-153. 10.1016/j.biosystemseng.2022.06.016 . hal-03948428

HAL Id: hal-03948428

<https://univ-angers.hal.science/hal-03948428v1>

Submitted on 2 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RoseSegNet: An Attention-Based Deep Learning Architecture For Organ Segmentation of Plants

Kaya Turgut^a, Helin Dutagaci^a and David Rousseau^{b,*}

^a*Eskisehir Osmangazi University, Eskisehir, Turkey*

^b*Université d'Angers, Angers, France*

ARTICLE INFO

Keywords:

3D phenotyping
plant segmentation
point-based deep learning
self-attention
surface descriptor


ABSTRACT

An important component for the advancement of plant breeding, genetics, and genomics research is the rapid and accurate measurement of phenotypic traits of large plant populations. The phenotypic data that are of interest can be at multiple levels of plant organization including organ-level geometric characteristics as well as the spatial organization of the organs. 3D computer vision enabling 3D geometry acquisition and processing promises to supply fast, automated phenotypic data collection. One important component of the processing pipeline is the segmentation of the plant into its structural components, such as leaves, stems, and flowers. In this paper, a novel 3D point-based deep learning network, namely RoseSegNet, is proposed for segmentation of point clouds of rosebush plants to their organs. The network is equipped with two attention-based modules, one for extracting contextual features at the encoder phase, another for feature propagation at the decoder phase. The network processes regions of points in a hierarchical manner, where at each level, point features are aggregated using attention-based operators. The aggregation is performed by incorporating point relations both within and between the receptive fields, defined by the hierarchical organization of points. RoseSegNet outperforms the widely-used architecture PointNet++ by 4% in terms of *MIoU* on the publicly available ROSE-X data set. Also, it is demonstrated that introducing local surface features together with the spatial coordinates of each 3D point at the input level boosts the segmentation performance of both networks by 9% in terms of *MIoU*.

Nomenclature

$(\lambda_1, \lambda_2, \lambda_3)$	Eigenvalues of the covariance matrix Σ	\mathcal{F}	Set of point features
(F_1, F_2, F_3, F_4)	Local surface features	\mathcal{N}_s^*	Neighbourhood of point p_i
(x_i, y_i, z_i)	Coordinates of point p_i	\mathcal{P}	The set of points representing a point cloud
α	The weight function in the <i>Intra-Emb</i> operator	\mathcal{R}_i^{inter}	The set of nearest L centre points to c_i
β	The transformation function to obtain the query vector in the <i>Intra-Emb</i> operator	\mathcal{R}_i^{intra}	The point set falling inside the region centred at c_i
Δc_u	The relative position vector between p and c_u	\bar{p}	Mean of the coordinates of the points in \mathcal{N}_s
δ_k	The position vector of k^{th} point in the <i>Intra-Emb</i> operator	ϕ	The non-linear transformation in the <i>Intra-Emb</i> operator
γ	The non-linear transformation of α in the <i>Intra-Emb</i> operator	ψ	The transformation function to obtain value vectors in the <i>Intra-Emb</i> operator
$\hat{\alpha}$	The weight function in the <i>Att-Prop</i> module	ρ	Softmax function
$\hat{\beta}$	The transformation function to obtain the query vector in the <i>Att-Prop</i> module	Σ	Covariance matrix
$\hat{\gamma}$	The non-linear transformation of $\hat{\alpha}$ in the <i>Att-Prop</i> module	Θ	The linear transformation of δ_k in the <i>Intra-Emb</i> operator
$\hat{\phi}$	The non-linear transformation in the <i>Att-Prop</i> module	φ	The transformation function to obtain key vectors in the <i>Intra-Emb</i> operator
$\hat{\psi}$	The transformation function to obtain value vectors in the <i>Att-Prop</i> module	ϑ	A one-layer MLP with leaky ReLU to transform f_k for g_{res}^{intra}
$\hat{\phi}$	The transformation function to obtain key vectors in the <i>Att-Prop</i> module	C	Number of semantic categories of plant organs
\hat{W}_{ke}	The weights of the function $\hat{\phi}$	c_i	i^{th} centre point in set C
\hat{W}_{qu}	The weights of the function $\hat{\beta}$	c_u	u^{th} nearest centre point to p in the <i>Att-Prop</i> module
\hat{W}_{va}	The weights of the function $\hat{\psi}$	D	Dimension of features in \mathcal{F}
C	Set of centre points	f_i	Features of point p_i
		f_k	Features of point p_k

*Corresponding author

 david.rousseau@univ-angers.fr (D. Rousseau)

ORCID(s):

f_{att}	The attention-based interpolated features in the <i>Att-Prop</i> module	R	Recall
f_{prop}	The propagated features	r_F	Radius of the neighbourhood for the extraction of local surface features
FN	Number of false negatives	r_T	Radius of the region centred at c_i
FP	Number of false positives	s	Factor of dimension reduction for input features to attention-based modules
g	Features of point c_i	TP	Number of true positives
g_u	Features of point c_u	U	The number of closest centre points in the <i>Att-Prop</i> module
g_{att}^{inter}	Attention-based aggregated features in the <i>Inter-Emb</i> operator	W_{ke}	The weights of the function φ
g_{att}^{intra}	Attention-based aggregated features in the <i>Intra-Emb</i> operator	W_{qu}	The weights of the function β
$g_{context}$	Concatenated features of <i>Intra-Emb</i> and <i>Inter-Emb</i> operators	W_{va}	The weights of the function ψ
g_{inter}	The output features of the <i>Inter-Emb</i> operator	<i>Att-Abs</i>	Attention-based abstraction
g_{intra}	The output features of the <i>Intra-Emb</i> operator	<i>Att-Prop</i>	Attention-based propagation
g_{res}^{inter}	The residual features in the <i>Inter-Emb</i> operator	<i>Inter-Emb</i>	Inter-region Embedding
g_{res}^{intra}	The residual features in the <i>Intra-Emb</i> operator	<i>Intra-Emb</i>	Intra-region Embedding
IoU	Intersection over Union	CNN	Convolutional Neural Networks
K	Number of points randomly selected from \mathcal{R}_i^{intra}	FPFH	Fast Point Feature Histograms
L	Number of centre points in \mathcal{R}_i^{inter}	FPS	Farthest point sampling
M	Number of points in set C	LFPC-s	Local Features on Point Cloud - supervised
$MIoU$	Mean Intersection over Union	LFPC-u	Local Features on Point Cloud - unsupervised
N	Number of points in \mathcal{P}	NLP	Natural Language Processing
N_0	Number of points in the input point cloud	PCA	Principal Component Analysis
P	Precision	SVM	Support Vector Machines
p_i	i^{th} 3D point in \mathcal{P}	VCNN	Voxel-based convolutional neural networks
p_j	j^{th} neighbour of point p_i		
p_k	k^{th} 3D point in \mathcal{R}_i^{intra}		

1. Introduction

Component phenotypes of plants refer to measurements of individual components of plants such as leaves, branches and flowers (Choudhury et al., 2019). Leaf area, branching angle, stem length are examples to such measurements, which are traditionally obtained through manual methods. Manual phenotyping, being highly labor-intensive and error-prone, is far from meeting the demand for rapid phenotyping of large populations of plants to analyse complex interactions between genotypes and the environment (Minervini et al., 2015). Automated phenotyping through computer vision and machine learning techniques has been intensively pursued in the last decade in order to break this phenotyping bottleneck (Mochida et al., 2018).

The visual plant data can be digitised through 2D photographic imaging (Zhang et al., 2021; Xu et al., 2019; Feldmann & Tabb, 2022) or through 3D sensing (Liu et al., 2020b; Bao et al., 2019). In either case, component phenotyping requires segmentation of the visual plant data into individual organs, such as leaves, stems, nodes, fruits, and flowers. The accurate segmentation of plants to the organs is also critical to extract morphological and architectural traits for automated high-throughput phenotyping.

2D image-based systems are widely used for segmentation of plant parts and estimation of organ-level traits. However, 2D photographic imaging poses challenges such as self-occlusion, missing data, and the variability due to illumination conditions. The lack of 3D depth information complicates the accurate evaluation of many traits such as component size, shape, orientation, and location. The geometrical data of plants in the form of point clouds, depth

18 images, etc, acquired through 3D sensors, supplies direct access to such measurements, provided that individual organs
19 are accurately segmented.

20 The general practise for segmentation of organs from 3D plant models has been the extraction of local surface
21 features that describe the local geometric information around each 3D point. Local features capture distinguishing
22 properties of organ classes and model within class variability (Ziamtsov & Navlakha, 2019). Examples to such local
23 surface features are first and second tensor features (Elnashef et al., 2019), Fast Point Feature Histograms (FPFH)
24 (Wahabzada et al., 2015; Sodhi et al., 2017) and eigenvalues of local covariance matrix (Dey et al., 2012; Dutagaci
25 et al., 2020). Once these point-based features are extracted, the segmentation is performed through classifying each
26 point with traditional machine learning approaches, such as Support Vector Machines (SVM) or Random Forests.

27 Classification of handcrafted surface features at the local point level is effective to some degree; however, depends
28 heavily on the design of the features and is blind to the contextual information at larger scales. Deep neural networks
29 are capable of simultaneously extracting and aggregating features at multiple scales providing context information and
30 weighting relevant features according to a loss function evaluated on training data.

31 Point cloud data can be obtained through multi-view stereo or RGB-D image acquisition. In these cases, it is
32 possible to apply standard convolutional neural networks (CNN) to individual colour or depth images (Shi et al.,
33 2019; Liu et al., 2020a; Majeed et al., 2020). Point clouds can be the raw output of other acquisition devices such as
34 3D LiDARs. The irregular structure of point clouds poses a challenge for direct application of standard CNNs, which
35 require regular data grids of fixed-size as input. To overcome this limitation, rendering 3D point clouds onto 2D images
36 has been proposed (Japes et al., 2018; Jin et al., 2018; Wang et al., 2019). Segmentation is performed on the rendered 2D
37 images through convolutional neural networks such as U-net, Mask R-CNN, or Fast-RCNN. Then, segmented pixels
38 are associated with the original 3D point cloud data in accordance with the transformation relationship established
39 between the images and the point cloud. Although these methods enable the application of CNNs to data derived from
40 3D plant models, they do not operate directly on the point clouds in 3D space. Another issue is the computational
41 cost of the rendering and projection phases. As another strategy, the point cloud can be converted to a volumetric
42 form that preserves the spatial relationships in 3D and enable the application of 3D CNNs. Jin et al. (2020) and Le
43 Louëdec & Cielniak (2021) proposed voxel-based convolutional neural networks (VCNN) for maize stem and leaf
44 segmentation and segmentation of strawberry fruit, respectively. The disadvantage with volumetric approaches is the
45 trade-off between computational cost and resolution due to quantization.

46 Recent advances on extension of deep neural networks for direct application on 3D point cloud data are key to
47 exploring their capabilities for 3D plant analysis. Such advances on 3D point-based deep neural networks started with
48 the introduction of PointNet (Qi et al., 2017a) and its local variant PointNet++ (Qi et al., 2017b), and exploded in
49 the last decade (Guo et al., 2021b). Despite this proliferation, the application of 3D point-based networks on plant

50 phenotyping is limited to a few studies, mainly due to the scarcity of annotated 3D plant data sets (Chaudhury et al.,
51 2020). In work (Turgut et al., 2022), point-based deep learning architectures were compared for organ segmentation
52 on ROSE-X data set (Dutagaci et al., 2020) and their performances were enhanced with the incorporation of synthetic
53 point cloud data. Schunck et al. (2021) released a multi-temporal data set and provided baseline results of point-based
54 deep learning networks such as PointNet, PointNet++ and LatticeNet (Rosu et al., 2020). Chaudhury et al. (2021)
55 explored the performance of PointNet++ model trained with virtual plants on real plants and obtained promising
56 results. Boogaard et al. (2021) demonstrated the ability of PointNet++ of segmenting incomplete point clouds of
57 cucumber plants and also showed that spectral information boosted the performance. Morel et al. (2020) proposed a
58 network based on PointNet and PointNet++ to segment virtual trees into woody and leaf parts. After the point cloud was
59 partitioned into overlapping sub-clouds, a variant of PointNet++ was used to extract the global information of each sub-
60 cloud. The global feature and local features extracted by Principal Component Analysis (PCA) were concatenated and
61 PointNet was applied recursively to predict labels. Ghahremani et al. (2021) introduced Pattern-Net to segment wheat
62 models into organs. The point cloud was decomposed into multiple and different subsets via a random downsampling
63 operator and a feature extraction pattern was applied across all subsets to extract the stationary patterns.

64 The problems associated with sampling and organizing 3D points within the structure of neural networks for
65 effective surface characterization are still not solved. Morel et al. (2020) showed that introducing geometric local
66 descriptors as input to 3D deep learning networks provides additional information on the distribution of points
67 in the local neighbourhood and enhances the performance of the classifier significantly for segmentation of trees.
68 Boogaard et al. (2021) used the spectral information as additional feature channels for improvement of the classification
69 performance of the deep learning architecture. The addition of prior information of plant organs formulated as local
70 geometric and spectral characteristics provides a significant benefit to the deep learning architectures to learn the
71 underlying latent surface information.

72 Recently, due to the success of transformer networks in the Natural Language Processing (NLP) domain (Vaswani
73 et al., 2017), the concept of self-attention was adapted to different domains to reveal contextual information present
74 across longer ranges (Khan et al., 2021; Wang et al., 2021). The self-attention mechanism is well-suited to extract
75 latent relationships of points in the 3D model analysis as it is inherently permutation-invariant for processing point
76 cloud data (Guo et al., 2021a). Zhao et al. (2021) showed that self-attention can extract descriptive features considering
77 the correlation between points. To the best of our knowledge, the potential of attention mechanism integrated into a
78 point-based deep learning architecture has not been previously explored for 3D plant analysis.

79 In this work, a novel point-based deep network architecture, which is called RoseSegNet, is proposed to segment 3D
80 plant models into structural parts. The network is equipped with self-attention mechanisms. It is composed of encoder
81 and decoder parts, each designed in a hierarchical manner. The attention-based modules embedded into the layers of

82 the encoder structure, abstract point features by relating them both within local regions (intra-region), and between
83 representatives of local regions (inter-region). Residual connections are inserted to both the attention-based intra-
84 region and inter-region embedding operators. In the decoder part, attention-based propagation modules are employed
85 to hierarchically interpolate abstracted point features back to the input point cloud. The network is trained and tested
86 on the fully annotated 3D rosebush models in the ROSE-X data set (Dutagaci et al., 2020) to segment them into
87 flower, leaf, and stem parts. The results demonstrate that RoseSegNet outperforms the widely-used deep learning
88 architecture, PointNet++. Moreover, introducing local surface descriptors as input to both networks results in a boost
89 in segmentation performance.

90 The contributions of this work can be summarised as follows:

- 91 • A novel 3D deep learning network for efficient and accurate segmentation of 3D plant models is introduced. The
92 network is equipped with self-attention mechanisms and residual connections to model interactions among local
93 structures.
- 94 • It is demonstrated that, in their current state, 3D point-based deep neural networks benefit from augmenting the
95 point coordinates with local surface features at the input stage.

96 **2. Material and methods**

97 In Section 2.1 the data set is described and details on the steps of data pre-processing are given. In Section 2.2, the
98 local surface descriptors that are provided as input features to the networks are defined. In Section 2.3, the operations
99 of the proposed attention-based abstraction and propagation modules are explained. Lastly, RoseSegNet is introduced
100 as the full attention-based hierarchical point processing network in Section 2.4

101 **2.1. Data set**

102 The publicly available ROSE-X data set (Dutagaci et al., 2020) is used to test the proposed deep learning
103 architecture and to analyse the effect of incorporating local surface features. ROSE-X data set consists of 11 complete
104 3D point cloud models of real rosebush plants. The 3D data was acquired through X-ray tomography, and initially
105 modeled in volumetric form. The models were fully annotated at point level, into three classes as flower, leaf, and stem.
106 The stem class includes the main stem, the branches, and the petioles. Two point cloud samples from the ROSE-X data
107 set are given in Fig. 1.

108 The models in the ROSE-X data set are provided in the following forms: (1) raw X-ray image stacks, (2) labeled
109 binary volume masks, (3) labeled binary volume masks indicating the voxels only on the surfaces of the plant shoots,
110 (4) labeled point clouds, (5) labeled point clouds composed of the points on the surfaces of the plant shoots. In this
111 work, point clouds that represent the surfaces of the plant shoots are used. More details related to the data set are given

112 in (Dutagaci et al., 2020). The information on file formats are also explained in the supplementary material available
 113 at the publisher site (Dutagaci et al., 2020).

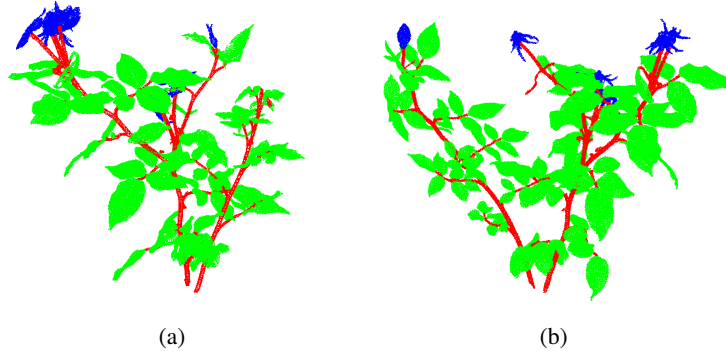


Figure 1: Two samples from ROSE-X data set. ROSE-X data set consists of 11 complete 3D point cloud models of real rosebush plants (Dutagaci et al., 2020). The models were fully annotated at point level, into three classes as flower, leaf, and stem. The stem class includes the main stem, the branches, and the petioles.

114 2.1.1. Data Pre-processing

115 3D point-based deep learning networks accept a fixed number of points as input. This fixed number is denoted as N_0 ,
 116 the number of points in the point cloud provided as input to the network. For large point clouds, subsampling the entire
 117 data to the required size is not an option since it would result in a significant loss of geometric information. The practise
 118 is to partition large point clouds into blocks of predetermined size. The set of points in a block is then processed as an
 119 independent point cloud by the network, both at the training and test phases. The off-line data preparation procedure
 120 described in the work of Li et al. (2018) is followed. Each point cloud representing a complete rosebush model is
 121 divided into non-overlapping blocks.

122 The first stage is the partitioning of the point cloud into fixed-size non-overlapping blocks. If the number of points
 123 in a block is less than 10% of the predetermined number of points (N_0), the block is merged into a neighbouring block.
 124 The second stage forces the point distribution of each block to be homogeneous. Each block is divided into voxels with
 125 predefined size and the average number of points over all the voxels is calculated. For voxels that have points below
 126 the average value, the number of points is duplicated to obtain regularly distributed blocks. The last stage ensures that
 127 the number of points in each block is equal to N_0 . If the number of points in a block are higher than N_0 , the block
 128 is separated into new blocks that define the same region with differently sampled points. If the number of points in a
 129 block are less than N_0 , random points are duplicated such that the number of points is raised to N_0 .

130 At the training phase, block partitioning over a full rosebush model is performed with two different offset values
 131 (0 and 5cm), thus two sets of blocks are extracted from each model. This operation both provides augmented data
 132 and pushes the discontinuities at the blocks in the first set towards the centres of the blocks in the second set. At the

133 inference phase, the same offset values are applied to obtain two sets of blocks from a rosebush model. Each block is
 134 processed independently by the network, which produces the class probabilities of the points in the block as the output.
 135 For a single point in the rosebush model, class probabilities of the points are calculated by the network separately for
 136 each block. The final class of a point is determined according to the highest value among the probability scores of the
 137 corresponding points contained in two overlapping blocks.

138 The choice of the block size depends on the resolution of the point cloud. A large block size results in a significant
 139 loss of geometric detail due to subsampling, while small blocks lack contextual information among neighbouring plant
 140 organs. In the experiments, the block size is set as $10cm$, which provides a good compromise between point resolution
 141 and context range. Examples to extracted blocks are given in Fig. 2. The grid size, and the number of final points in
 142 each block are set as $0.2cm$, and 8192, respectively.

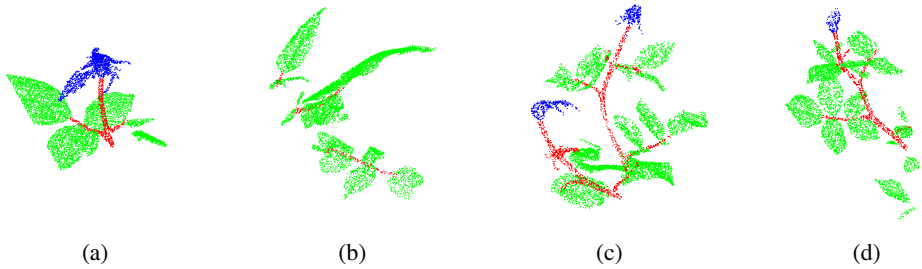


Figure 2: Examples of blocks from ROSE-X data set. Each point cloud representing a complete rosebush model is divided into non-overlapping blocks. The set of points in a block is then processed as an independent point cloud by the network, both at the training and test phases. In the experiments, the block size is set as $10cm$, which provides a good compromise between point resolution and context range.

143 2.2. Local Surface Features

144 3D point-based networks are expected to take as input the raw 3D coordinates of points, occasionally together
 145 with surface normals, and to probe and encode class-specific properties out of this raw data. However, at their current
 146 stage, the 3D point-based networks are still progressing in their manner of organizing the geometric data at the local
 147 level. To enhance the informative power of the raw point coordinates, hand-crafted point-based local surface features
 148 can be incorporated as input attributes. Then, the networks can exploit this additional information for encoding and
 149 aggregation of the interactions of local structures at various scales.

150 The eigenvalues of the local point covariance matrix are used to define the surface features. Let the input point
 151 cloud be denoted as the set $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ where the each point $p_i = (x_i, y_i, z_i)$ is represented in 3D coordinates.
 152 The neighbourhood of a point p_i can be defined as $\mathcal{N}_i = \{p_j : \|p_i - p_j\| < r_F\}$, where r_F is the radius of the
 153 neighbourhood. The covariance matrix of the points in the neighbourhood is calculated as:

$$\Sigma = \frac{1}{\mathcal{N}_i - 1} \sum_{p_j \in \mathcal{N}_i} (p_j - \bar{p}_i)(p_j - \bar{p}_i)^T \quad (1)$$

154 where \bar{p}_i denotes the mean of the coordinate vectors of the points in the neighbourhood.

155 The eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$ of the covariance matrix Σ represent the amount of the variation of the points in
 156 the neighbourhood along three principal axes. They carry information about the local shape around the point p_i . For
 157 example, when λ_1 and λ_2 are close to zero and λ_3 is relatively large, that is indicative of an elongated, line-like structure.
 158 On a locally planar region, both λ_2 and λ_3 are expected to be larger than λ_1 . The relations between eigenvalues of
 159 the covariance matrix around each point can be used to distinguish line-like, plane-like, and spherical local structures,
 160 hence be used as local descriptors for classification of flower, leaf, and stem points. The following local features are
 161 used as given in the work of Dutagaci et al. (2020):

$$F_1 = \frac{\lambda_1}{\sqrt{\lambda_2 \lambda_3}}, F_2 = \frac{\lambda_2}{\lambda_3}, F_3 = \frac{\lambda_1}{\sqrt{\lambda_1 \lambda_2 \lambda_3}}, F_4 = \frac{\lambda_1}{\lambda_2} \quad (2)$$

162 Local regions of different sizes instead of a fixed-size neighbourhood are processed to provide information for
 163 multiple scales. The local features, $\{F_1, F_2, F_3, F_4\}$, are extracted from local neighbourhoods of six different radii
 164 around each point, amounting to 24 local features. The radii are selected as r_F , as 2, 3, 4, 5, 6, and 7mm.

165 2.3. The Attention-based Modules of the Network

166 Before providing the full architecture for RoseSegNet, two core modules are described: Attention-based abstraction
 167 module (*Att-Abs*) at the encoder and Attention-based propagation module (*Att-Prop*) at the decoder. The Attention-
 168 based abstraction module (*Att-Abs*) is responsible for extracting contextual information for each local region, by
 169 considering the relations of points within the region and the relations of the representative point of the region with
 170 representative points of neighbouring local regions. Residual connections are present in *Att-Abs* module in order to
 171 capture dominant features as well as contextual features. The Attention-based propagation module (*Att-Prop*) allows
 172 the contextual information to impact the rate of feature propagation at the decoder.

173 2.3.1. Attention-based Abstraction Module (*Att-Abs*)

174 The input to the *Att-Abs* module is composed of the spatial coordinates $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ with $p_i \in \mathbb{R}^3$ and
 175 features $\mathcal{F} = \{f_1, f_2, \dots, f_N\}$ with $f_i \in \mathbb{R}^D$ of the unordered point set, where N denotes the cardinality, and D
 176 denotes the dimension of input point features. The module extracts contextual features from the point set by relating
 177 points within regions around representative (centre) points and between these representative points. To this end, the *Att-*
 178 *Abs* module engages two operators: 1-) Intra-region Embedding (*Intra-Emb*) Operator and 2-) Inter-region Embedding
 179 (*Inter-Emb*) Operator.

180 The input point set is sampled with iterative farthest point sampling (FPS) algorithm to determine representative
 181 points $C = \{c_1, c_2, \dots, c_M\}$, where $M < N$. In the illustration given in Fig. 3, the black points represent the input points,
 182 which correspond to the aggregated points from the previous layer. The representative points at the current layer onto
 183 which features will be aggregated are depicted in red colour. Around each representative point, a region whose radius
 184 is fixed at the particular layer is defined. The *Intra-Emb* operator relates each centre point to the points within its region
 185 through an attention-based approach (Fig. 3b). The *Inter-Emb* operator embeds longer range interactions by relating
 186 each centre point to other centre points of neighbouring regions (Fig. 3c). The *Att-Abs* module combines within-region
 187 and between-region contextual features extracted via the two operators.

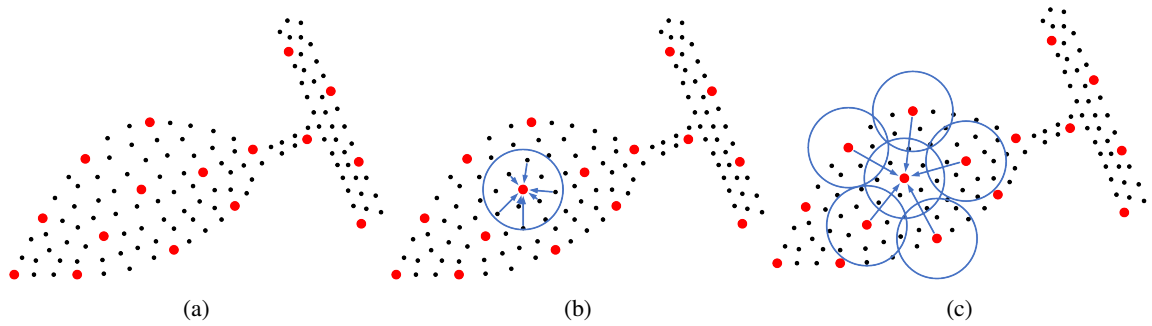


Figure 3: Point sampling and feature aggregation: a-) Red points are sampled using farthest point sampling (FPS) algorithm from the aggregated points (black points) of the previous layer. Each red point defines a region of fixed radius at the particular layer. b-) Intra-region interactions. K points are randomly selected. c-) Inter-region interactions. L nearest centre points are selected.

188 The diagram of the *Att-Abs* module is given in Fig. 4. After centre points are determined by the FPS algorithm, the
 189 grouping stage of the intra-region points and neighbourhoods of centre points for *Intra-Emb* and *Inter-Emb* operators
 190 are carried out in parallel. The grouped points together with their features are fed to *Intra-Emb* and *Inter-Emb* operators.
 191 Finally, context-aware features returned by the two operators are concatenated. In the following subsections, details of
 192 the *Intra-Emb* and *Inter-Emb* operators are given.

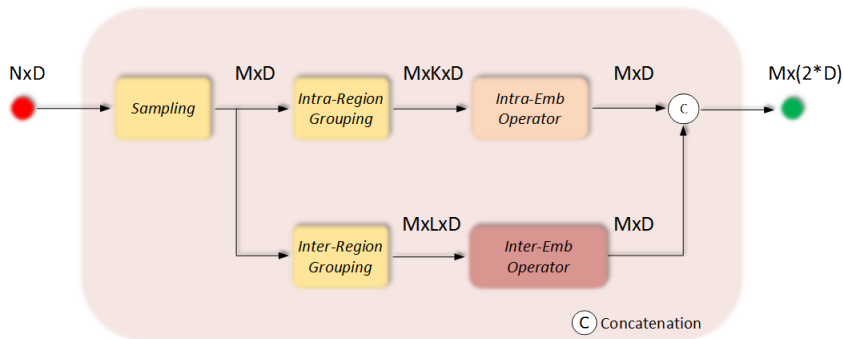


Figure 4: The attention-based abstraction (*Att-Abs*) module. The module is responsible for extracting contextual information for each local region.

193 *Intra-Emb Operator*: The grouping stage prior to *Intra-Emb* operation corresponds to determining the points in
 194 the region centred at the representative point c_i . This region of radius r_T is called the receptive field centred at
 195 c_i at the current layer. K points are randomly selected among the points falling inside this region to form the set
 196 $\mathcal{R}_i^{intra} = \{p_{i,1}, p_{i,2}, \dots, p_{i,K}\}$ (Fig. 3b). For simplicity, the index i is dropped, and the selected points within the region
 197 are denoted as p_k , and the corresponding input features as f_k , with $k = 1, \dots, K$. The input feature vector of the centre
 198 point is denoted as g .

199 With the attention-based approaches, query, key, and value vectors are obtained via transformations of the features
 200 of entities whose relationships are to be revealed (Vaswani et al., 2017). The goal of the *Intra-Emb* operation is to
 201 encode the relationship between the centre point and the points in the region defined by the centre point. A point
 202 feature aggregation approach similar to the point transformer described in the work of Zhao et al. (2021) is followed.
 203 The query vector is set to be a transformation of the features of the centre point, as '*query*' : $\beta(g, W_{qu})$. The key and
 204 value vectors are transformations of the features of K points within the region, and are calculated as '*key*' : $\varphi(f_k, W_{ke})$
 205 and '*value*' : $\psi(f_k, W_{va})$. The functions $\beta, \varphi, \psi : \mathbb{R}^D \rightarrow \mathbb{R}^{D/s}$ map the input features linearly to lower dimensions
 206 through the transformations W_{qu}, W_{ke} , and W_{va} , respectively, which are to be learned through training. For all attention-
 207 based modules in the network, s is set to 2. For the sake of simplicity, the transformation parameters from the arguments
 208 are dropped, and the transformed features are denoted as '*query*' : $\beta(g)$, '*key*' : $\varphi(f_k)$, and '*value*' : $\psi(f_k)$.

209 The block diagram of the *Intra-Emb* operation is given in Fig. 5. The features aggregated on the centre point through
 210 attention mechanism are calculated as:

$$g_{att}^{intra} = \phi \left(\sum_{k=1}^K \alpha(g, f_k, \delta_k) \odot (\psi(f_k) + \delta_k) \right) \quad (3)$$

211 where \odot is the Hamadard product, α is the weight function, δ is the position encoded vector, ϕ is a non-linear
 212 transformation function and K is the number of points sampled from the local region. The transformation function
 213 ϕ is used to increase the feature dimension of the aggregated contextual feature back to the original feature size. The
 214 weight function α measures and transforms the dissimilarity between the transformed features ('*query*' vector) of the
 215 centre point and transformed features ('*key*' vectors) of the K points within the region. The function also incorporates
 216 a transformation of Euclidean distance vectors between the centre point and the K points, as positional encoding. The
 217 aggregation is then performed by weighing the transformed features (corresponding to '*value*' vectors) of K points.
 218 The weight function is defined as:

$$\alpha(g, f_k, \delta_k) = \rho \left(\gamma(\beta(g) - \varphi(f_k) + \delta_k) \right) . \quad (4)$$

219 The relation between 'query' $\beta(g)$ and 'key' $\varphi(f_k)$ vectors is represented by the subtraction operation. The position
 220 encoded vector δ_k is a linear function of the relative position of the centre point to the k^{th} point in the region:

$$\delta_k = \Theta(c - p_k). \quad (5)$$

221 The parameters of the linear transformation function Θ is learned through training. The position encoded vector is
 222 added both to the difference of query $\beta(g)$ and key $\varphi(f_k)$ vectors in the weight function and the value vectors $\psi(f_k)$
 223 in the aggregation function.

224 The function γ is a two-layered network, where the first layer is nonlinear and the second layer is linear. It is used
 225 to learn the embedding that will effectively represent the relative dissimilarity measures between points. The softmax
 226 function ρ is used to normalise the weights across K 'value' vectors.

227 Inspired by the effectiveness of residual networks (He et al., 2016), the max-pooled version (g_{res}^{intra}) of the
 228 transformed features $\{\vartheta(f_k)\}$ of K points are added to the attention-based aggregated features (g_{att}^{intra}). The objective
 229 here is to let the dominant feature among the transformed input features of the K points contribute to the aggregated
 230 output:

$$g_{intra} = g_{att}^{intra} + g_{res}^{intra} \quad (6)$$

$$g_{res}^{intra} = \max_{k=1, \dots, K} \vartheta(f_k) \quad (7)$$

231 where ϑ is a one-layer MLP with leaky ReLU.

232 Inter-Emb Operator: While the *Intra-Emb* operator aggregates features of K points within the receptive field
 233 defined around each centre point (Fig. 3b), the *Inter-Emb* operator explores the interactions between receptive fields
 234 through relating features of each centre point to its neighbouring centre points (Fig. 3c). The two operators run in
 235 parallel.

236 The *Inter-Emb* operator is designed in the same manner as the *Intra-Emb*. In *Inter-Emb*, the attention mechanism
 237 operates on the input features of a target centre point ('query') and other centre points ('keys' and 'values') in its
 238 vicinity. The *Inter-Emb* operator aggregates point features in longer ranges as compared to the *Intra-Emb* operator.

239 In *Inter-Emb*, for each centre point c_i , the nearest neighbour search algorithm is applied to find L closest centre
 240 points to form the group $R_i^{inter} = \{c_{i,1}, c_{i,2}, \dots, c_{i,L}\}$. The 'query' vector is a transformation of the features of the centre
 241 point. The key and value vectors are calculated as the transformed features of the neighbouring L centre points.

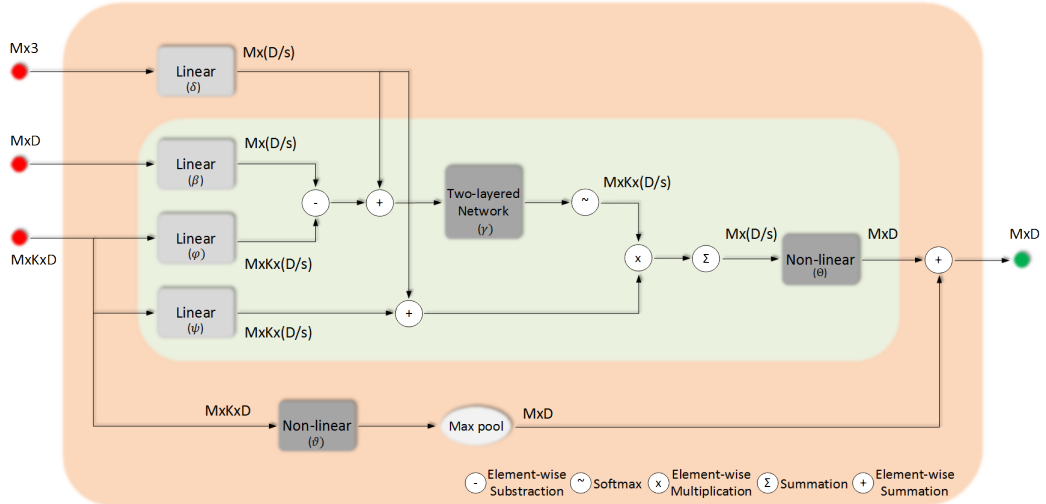


Figure 5: *Intra-Emb* operator. The operator relates each centre point to the points within its region through an attention-based approach

242 Through attention-based feature aggregation, the inter-region features g_{att}^{inter} are obtained. The output of the *Inter-*
 243 *Emb* operator is

$$g_{inter} = g_{att}^{inter} + g_{res}^{inter} \quad (8)$$

244 where g_{res}^{inter} represents residual max-pooled features. The details for the computation of g_{att}^{inter} and g_{res}^{inter} can be found
 245 in the Supplementary Material.

246 As stated before, the features extracted by the intra-region and inter-region operators are concatenated to form the
 247 output contextual feature of the Attention-based Abstraction (*Att-Abs*) Module:

$$g_{context} = g_{intra} \oplus g_{inter} \quad (9)$$

248 where \oplus denotes concatenation operation.

249 2.3.2. Attention-based Propagation Module (*Att-Prop*)

250 The Attention-based propagation (*Att-Prop*) module operates at the decoder layers, which propagate the aggregated
 251 features back to the original points. The *Att-Prop* module allows the contextual information to impact the rate of feature
 252 propagation. The block diagram of the *Att-Prop* module is given in Fig. 6.

253 At the encoder, each successive *Att-Abs* module yields more descriptive and long-ranged features, but the features
 254 are aggregated at fewer points. The general practise for distributing the features aggregated at various layers back to the
 255 original point cloud is to perform distance-based interpolation and to introduce skip links from the abstraction layers

256 to the propagation layers (Qi et al., 2017b). In this work, the use of self-attention is proposed to learn the interpolation
257 weights according to the relation of point coordinates and features.

258 Recall that the *Att-Abs* module at the encoder accepts a point set (\mathcal{P}) of size N with 3D coordinates and point-
259 features and returns a subset (\mathcal{C}) of size M , with output features aggregated through attention mechanisms. The
260 representative (centre) points in \mathcal{C} are determined by Furthest Point Sampling. The propagation at the decoder stage
261 aims to distribute the aggregated features of the set \mathcal{C} to the points in set \mathcal{P} . The *Att-Prop* module relates each point
262 in \mathcal{P} to its neighbours in the set \mathcal{C} through an attention mechanism. Given a point coordinate $p \in \mathcal{P}$, its U nearest
263 neighbours among the centre points are determined as $\{c_1, c_2, \dots, c_U\} \subset \mathcal{C}$, with corresponding features as $g_u \in \mathbb{R}^{D_1}$,
264 $u = 1, \dots, U$. The query vector is a linear transformation of the relative position vector between point p and centre point
265 c_u , defined as 'query' : $\hat{\beta}(\Delta c_u, \hat{W}_{qu})$, where $\Delta c_u = p - c_u$. The key and value vectors are computed through linear
266 transformations of the features g_u as 'key' : $\hat{\phi}(g_u, \hat{W}_{ke})$ and 'value' : $\hat{\psi}(g_u, \hat{W}_{va})$.

267 The feature for point p is interpolated from the U points through the following attention-based weighting scheme:

$$f_{att} = \hat{\phi} \left(\sum_{u=1}^U \hat{\alpha}(\Delta c_u, g_u) \odot \hat{\psi}(g_u) \right) \quad (10)$$

268 where \odot is the Hamadard product, $\hat{\alpha}$ is the weight function, and $\hat{\phi}$ is a non-linear function that transforms the features
269 into their original dimensionality.

270 The weight vector is determined as:

$$\hat{\alpha}(\Delta c_u, g_u) = \rho \left(\hat{\gamma}(\hat{\beta}(\Delta c_u) - \hat{\phi}(g_u)) \right) \quad (11)$$

271 where $\hat{\gamma}$ is a two-layered network and ρ is the softmax function. Weight vector $\hat{\alpha}(\Delta c_u, g_u)$ is learned according to the
272 relation function which is a subtraction of linear transformations of relative point coordinates and features.

273 Given that the features of point p were determined as $f \in \mathbb{R}^{D_0}$ at the encoder stage, the final propagated point
274 features $f_{prop} \in \mathbb{R}^{D_0+D_1}$ at the decoder stage is set as the concatenation of f and the interpolated features $f_{att} \in \mathbb{R}^{D_1}$:

$$f_{prop} = f_{att} \oplus f \quad (12)$$

275 The features f are provided to the decoder through skip links from the encoder.

276 2.4. The Network Architecture

277 The proposed deep learning architecture, RoseSegNet, for organ segmentation of plants is given in Fig. 7. The
278 network is built on an encoder-decoder structure. The encoder follows a local-to-global strategy, employing the *Att-*
279 *Abs* module at each layer to extract semantic affinities between points at the given scale of spatial interaction range.

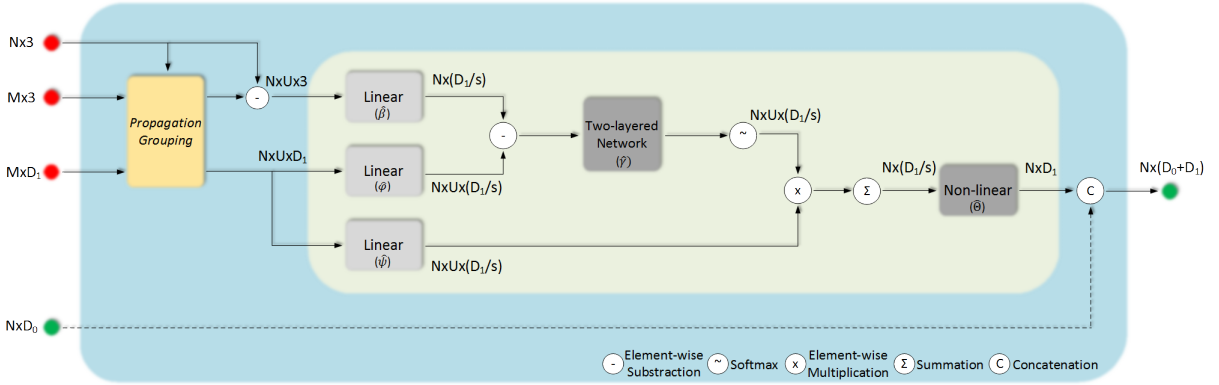


Figure 6: The attention-based propagation (*Att-Prop*) module. The module allows the contextual information to impact the rate of feature propagation at decoder layers.

280 By this strategy, the receptive fields are gradually expanded, and the spatial interaction ranges between aggregated
 281 points become longer. Also, the number of regions processed by successive abstraction modules is decreased to mimic
 282 convolution neural networks. The decoder is responsible to propagate aggregated features at successive layers of the
 283 encoder back to the original points in an hierarchical manner through *Att-Prop* modules. This allows that each point
 284 in the original set is enriched by the features carrying context information from various scales. The semantic labels of
 285 the points are then inferred through these informative point features.

286 Unit embedding operators are used before each *Att-Abs* module at the encoder stage, and after each *Att-Prop*
 287 module at the decoder stage. These operators consist of weight-shared MLPs which uplift the input features to higher
 288 dimensions to enrich their representation power at the encoder stage, and decrease the dimensionality at the decoder
 289 stage.

290 In the encoder part, four layers, each equipped with *Att-Abs* modules are used to aggregate features with a local-
 291 to-global strategy. The input point set with $N_0 = 8192$ points is downsampled to 1024, 256, 64, and 16 points through
 292 these four layers, and the receptive fields are expanded to $5mm$, $10mm$, $20mm$, and $40mm$, respectively. The number
 293 of points randomly sampled from intra-regions is set to $K = 32$ for the *Intra-Emb* operator. The number of nearest
 294 centre points is set to $L = 8$ for the *Inter-Emb* operator. The unit embedding operators at the four successive layers map
 295 feature dimensions to 64, 128, 256, and 512, respectively. The output dimension of each *Att-Abs* module is doubled
 296 since the embedded features on intra-region and inter-region are concatenated.

297 At the decoder stage, the contextual and long-range features of the down-sampled points are propagated to the
 298 original points. The decoder has four layers consisting of unit embedding operators and *Att-Prop* modules. Features of
 299 centre points are propagated to the point sets of the previous layers by using $U = 3$ nearest neighbours in the *Att-Prop*
 300 module. The output of the *Att-Prop* is the concatenation of the propagated features and the features of *Att-Abs* provided

301 by skip links. The unit embedding operators, inserted after the *Att-Prop* modules, map feature dimensions to 512, 256,
 302 128, and 64 at the four successive layers.

303 After the decoder stage, two fully connected layers with feature dimensions, 64 and C are engaged to extract scores
 304 of C categories for each point in the input point cloud.

305 All non-linear layers in the architecture include leaky ReLU activation function and batch normalization. Drop-out
 306 with keep ratio 0.5 is used on the last fully connected layer.

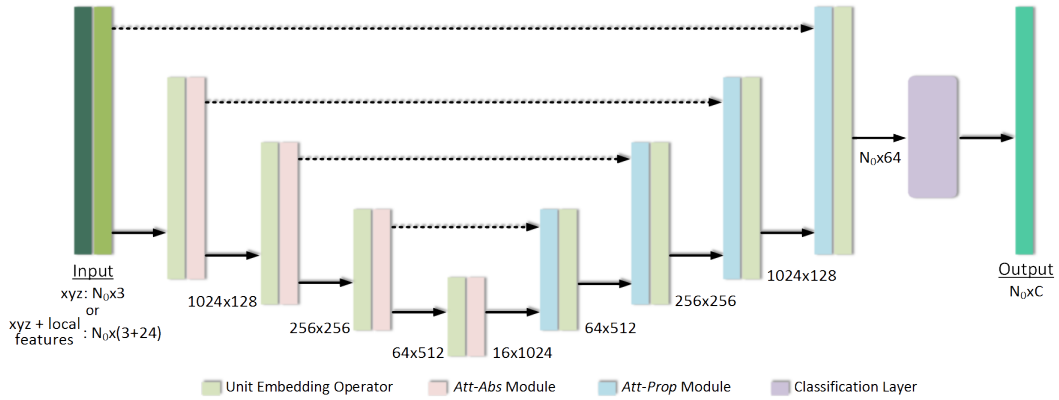


Figure 7: The architecture of RoseSegNet. The input is a point cloud of N_0 points. The point features at the input level can either be the 3D coordinates only (xyz) or the 3D coordinates and 24 local features together ($xyz + \text{local}$).

307 2.5. Implementation details

308 The network is implemented in Tensorflow, and trained on an NVIDIA Quadro P5000. The size of the input point
 309 cloud is set to $N_0 = 8192$ points. The training is performed using Adam optimiser, to minimise a weighted cross-
 310 entropy loss function. The batch size is 16, and training is completed after 200 epochs. The initial learning rate is set
 311 as 0.005. The learning rate is decayed by 0.7 for every 30 epochs. As data augmentation, rotated versions of the point
 312 clouds are used. Specifically, random rotations around the upright axis is applied to the point clouds.

313 2.6. Experimental setup

314 To evaluate the segmentation performance of the proposed network, 5-fold cross-validation experiments were
 315 performed. For each fold, a single rosebush model from ROSE-X data set was reserved for optimizing the weights
 316 of the network, and the remaining 10 models are used as test data. The data processing steps were followed to partition
 317 the point cloud into blocks as described in Section 2.1.1. In Table 1, the total number of training and test blocks that
 318 were extracted from training and test rosebush models are given for each fold. The number of blocks varied depending
 319 on the size of the corresponding rosebush.

Table 1

Number of training and test blocks used in the experiments.

	# blocks for training	# blocks for test
Fold 1	99	677
Fold 2	53	723
Fold 3	103	673
Fold 4	56	720
Fold 5	89	687

320 2.7. Evaluation metrics

321 In this study, three metrics were used to compare the success of plant organ segmentation: 1-) Precision (P), 2-)
322 Recall (R), and 3-) Intersection over Union (IoU). These metrics are defined as:

$$P_i = \frac{TP_i}{TP_i + FN_i} \quad (13)$$

$$R_i = \frac{TP_i}{TP_i + FP_i} \quad (14)$$

$$IoU_i = \frac{TP_i}{TP_i + FN_i + FP_i} \quad (15)$$

323 where the TP_i , FP_i and FN_i represent the number of true positives, the number of false positives and the number of
324 false negatives of each class $i \in \{flower, leaf, stem\}$. The categorization of a point as a true positive or otherwise is
325 demonstrated in Fig. 8.

		Prediction by the network	
		Classified as class i	Classified to a category other than i
Ground Truth	Belongs to class i	True Positive (TP_i)	False Negative (FN_i)
	Does not belong to class i	False Positive (FP_i)	True Negative (TN_i)

Figure 8: Confusion matrix ($i \in \{flower, leaf, stem\}$). A point is a true positive if both its actual category and predicted category is i . It is a false negative if the point's actual category is i and the network classified it wrongly to another class. It is a false positive if the point does not belong to class i , but the network classified it as class i . If neither the actual class nor the predicted class of the point is i , then it is a true negative.

326 **3. Results**

327 The segmentation performance of RoseSegNet is evaluated on the publicly available ROSE-X data set (Dutagaci
328 et al., 2020). The effect of employing the attention-based modules are analysed through ablation studies. Also, the
329 positive contribution of feeding the network with hand-crafted local features is demonstrated.

330 **3.1. Semantic Segmentation of ROSE-X**

331 The segmentation results of the proposed network, RoseSegNet, are given in Table 2, in comparison with three
332 other segmentation methods. Each evaluation value in the Table 2 represents the average and standard deviation over
333 the 5-fold experiments. The state-of-the-art point-based deep learning network PointNet++ was selected as a baseline
334 for comparison. The default architecture of PointNet++ was used, but it was trained with the same hyperparameters
335 (radii of local regions, K , number of epochs, batch size and learning rate) as those of RoseSegNet. For both PointNet++
336 and RoseSegNet, two different networks are trained. The first network accepts only the 3D coordinates as input point
337 features. The second network is designed to accept the local surface features (see Section 2.2) together with the 3D
338 coordinates as input point features.

339 The segmentation results on the same data set of two other methods described in (Dutagaci et al., 2020) are also
340 given. These methods are the LFPC-u (Local Features on Point Cloud - unsupervised) and The LFPC-s (Local Features
341 on Point Cloud - supervised). The supervised method uses Support Vector Machines (SVM) as the machine learning
342 model. The segmentation results in Table 2 are given as reported in (Dutagaci et al., 2020). These two methods were
343 used to classify points only to two categories as 'stem' and 'leaf'. The points annotated as 'flower' were ignored.

344 As observed from Table 2, in terms of IoU , RoseSegNet fed with local surface features outperforms the other
345 methods over all categories. RoseSegNet fed with local surface features outperforms LFPC-s over all performance
346 metrics. The 0.5% drop in terms of 'leaf' precision as compared to LFPC-u is compensated by 3%, 17%, and 5%
347 increase in terms of 'leaf' recall, 'stem' precision, and 'stem' recall, respectively.

348 RoseSegNet outperforms PointNet++ by 4% in terms of $MIoU$, with or without the use of local features. When
349 both networks are augmented with local features, PointNet++ performs with a 1.6% higher 'flower' precision, and a
350 0.1% higher 'leaf' recall. These are compensated by RoseSegNet with 14% higher 'flower' recall and 1% higher 'leaf'
351 precision.

352 PointNet++, relying only on the spatial coordinates of the input points, returns lower performance figures for
353 leaf and stem categories as compared to LFPC-u and LFPC-s. Notice that these two methods are representatives of
354 the traditional unsupervised and supervised techniques for organ segmentation of plants. RoseSegNet, without the
355 local features, gives lower 'leaf' and 'stem' IoU measures compared to LFPC-s. Supplying local features as input to
356 the networks boosts the performance significantly for both PointNet++ and RoseSegNet. The performance increase

in terms of $MIoU$ is 9% for both networks, when the input data is enriched by local features. These observations demonstrate the importance of augmenting input 3D point coordinates with corresponding local surface features while training the point-based networks.

Table 2

The segmentation performance of RoseSegNet on ROSE-X data set in comparison with PointNet++, LFPC-u, and LFPC-s

		LFPC-u	LFPC-s	PointNet++		RoseSegNet	
				xyz	xyz+local features	xyz	xyz+local features
<i>Precision</i>	<i>Flower</i>	-	-	88.92±5.31	92.86±2.03	86.55±8.18	91.20±3.79
	<i>Leaf</i>	98.23±0.33	97.19±0.48	95.74±0.84	96.76±1.01	96.53±0.25	97.78±1.11
	<i>Stem</i>	75.01±9.76	83.67±4.88	77.25±3.85	90.48±1.93	79.90±5.20	91.96±0.46
<i>Recall</i>	<i>Flower</i>	-	-	61.11±10.10	64.67±10.39	73.52±7.36	79.05±12.03
	<i>Leaf</i>	95.74±1.74	97.79±0.46	97.11±0.77	98.77±0.29	97.04±0.32	98.67±0.34
	<i>Stem</i>	88.03±1.82	80.50±1.29	82.90±2.47	92.78±3.97	83.01±3.98	92.87±1.37
<i>IoU</i>	<i>Flower</i>	-	-	56.17±6.71	61.51±9.30	65.01±2.41	72.91±7.59
	<i>Leaf</i>	94.10±1.54	95.10±0.46	93.08±0.68	95.60±0.77	93.77±0.29	96.50±0.78
	<i>Stem</i>	67.96±8.18	69.57±3.87	66.63±3.54	84.52±3.68	68.41±2.81	85.90±1.11
<i>MIoU</i>	-	-	71.96±2.16	80.55±3.80	75.73±1.02	85.10±2.85	

In Fig 9, visual segmentation results provided by PointNet++ and RoseSegNet on six sample rosebush blocks are given. The first column represents the ground truth. The second and third columns give the results obtained with PointNet++ and RoseSegNet, respectively, with the use of spatial coordinates only. The fourth and fifth columns depict the segmentation of the blocks by PointNet++ and RoseSegNet, respectively, when both networks are fed with local surface features.

In Figs 9a, 9c, 9e and 9f, it can be observed that petioles are classified as leaf points by PointNet++ without the use of local features. The addition of local features alleviates this confusion for both PointNet++ and RoseSegNet. However, petiole-leaf distinction is best modeled by RoseSegNet augmented with local features. This success can be attributed to the attention-based mechanisms that extract contextual information at leaf-stem boundaries.

Another source of error is the misclassification of some flower regions as either leaf or stem points (Figs 9a, 9c, 9d, 9e, and 9f). The misclassification is most pronounced with PointNet++ trained without the local surface features. With the exception of the block in Fig 9f, RoseSegNet with local features captured the variations among the flower points most effectively. The gain with IoU for the flower class with RoseSegNet over PointNet++ is significant (11%) as can be observed from Table 2. For the case of the flower in Fig 9f, the addition of local features to RoseSegNet lead to confusion of petals with leaves. Despite this example, the IoU value for the flower class is significantly higher, in average, for RoseSegNet operating with local features (Table 2).

An interesting result arises in Fig. 9a. The stipules in the rosebush models were originally labeled as 'stem' in the ground-truth annotation of ROSE-X data set. The attention-based RoseSegNet tends to classify those stipule points at the extremities as flowers, which is coherent with the contextual relation that elongated and short flower parts tend to

Table 3

Ablation study on RoseSegNet. The first row gives IoU values of the default RoseSegNet that includes *Att-Abs* and *Att-Prop* modules. Each *Att-Abs* module consists of *Intra-Emb* and *Inter-Emb* operators with residual connections. Number of neighbouring centre points is selected as $L = 8$ in *Inter-Emb* operator. In *Att-Prop* module, U is set to 3. Remaining rows of the Table give IoU values with the setting changed as specified in the first column.

Model	IoU_{flower}	IoU_{leaf}	IoU_{stem}	$MIoU$
<i>RoseSegNet</i>	81.99	97.26	85.26	88.17
Architecture				
w/o <i>Att-Abs</i>	78.48	96.71	86.31	87.17
w/o <i>Att-Prop</i>	79.43	96.90	86.58	87.64
w/o <i>Att-Abs, Att-Prop</i>	64.59	95.81	86.86	82.42
Att-Abs Module				
w/o <i>Inter-Emb</i>	79.41	97.01	86.40	87.61
w/o <i>Intra-Emb</i>	71.55	96.02	83.12	83.57
w/o Residual	77.19	96.87	85.57	86.54
L=4	76.72	96.90	87.14	86.92
L=12	81.23	97.22	85.95	88.13
L=16	80.61	96.86	86.39	87.96
Att-Prop Module				
U=1	77.97	96.72	85.77	86.82
U=5	79.42	96.99	86.46	87.62
U=8	78.45	96.49	84.39	86.44

379 occur at the extremities, while stem class is long ranging and usually followed by leaf or flower points. The assignment
380 of stipules to the flower class also contributes to the lower 'flower' precision yielded by RoseSegNet.

381 3.2. Ablation Study

382 Ablation experiments on RoseSegNet were conducted to observe the influence of different settings. The version
383 of RoseSegNet where local features were introduced to the network as input in addition to the spatial coordinates was
384 studied. The results of these experiments are given in Table 3. The first row of the table indicates the classification
385 performance of the default RoseSegNet. Each model in the remaining rows was trained with the same hyper-paramaters
386 and the same settings, except the one specified in the first column of the corresponding row.

387 Three experiments were performed to analyse the effect of exclusion of attention-based modules. When *Att-Abs* was
388 removed, the feature aggregation operation was replaced by PointNet++'s default abstraction strategy (max-pooling of
389 embedded features). When *Att-Prop* was excluded, the feature propagation was implemented according to the distance-
390 based scheme of PointNet++. Using either *Att-Abs* or *Att-Prop* modules has individually increased $MIoU$ by about
391 5% as compared to the case where neither was included. There was about 1% drop in IoU for the stem class with the
392 inclusion of attention-based modules, however, the increase in IoU for the flower class was significant (Table 3). The
393 attention-based modules helped increase the performance for the difficult and rare 'flower' class.

394 The *Att-Abs* module was assessed in terms of the contributions of the *Intra-Emb* and *Inter-Emb* operators. While the
395 performance of the network without the features encoded by the *Inter-Emb* operator decreased slightly as compared to
396 the default RoseSegNet, the performance drop without the *Intra-Emb* operator was significant (Table 3). The *Intra-Emb*
397 operator is more essential since it aggregates features within receptive fields in the spirit of a convolution operation.

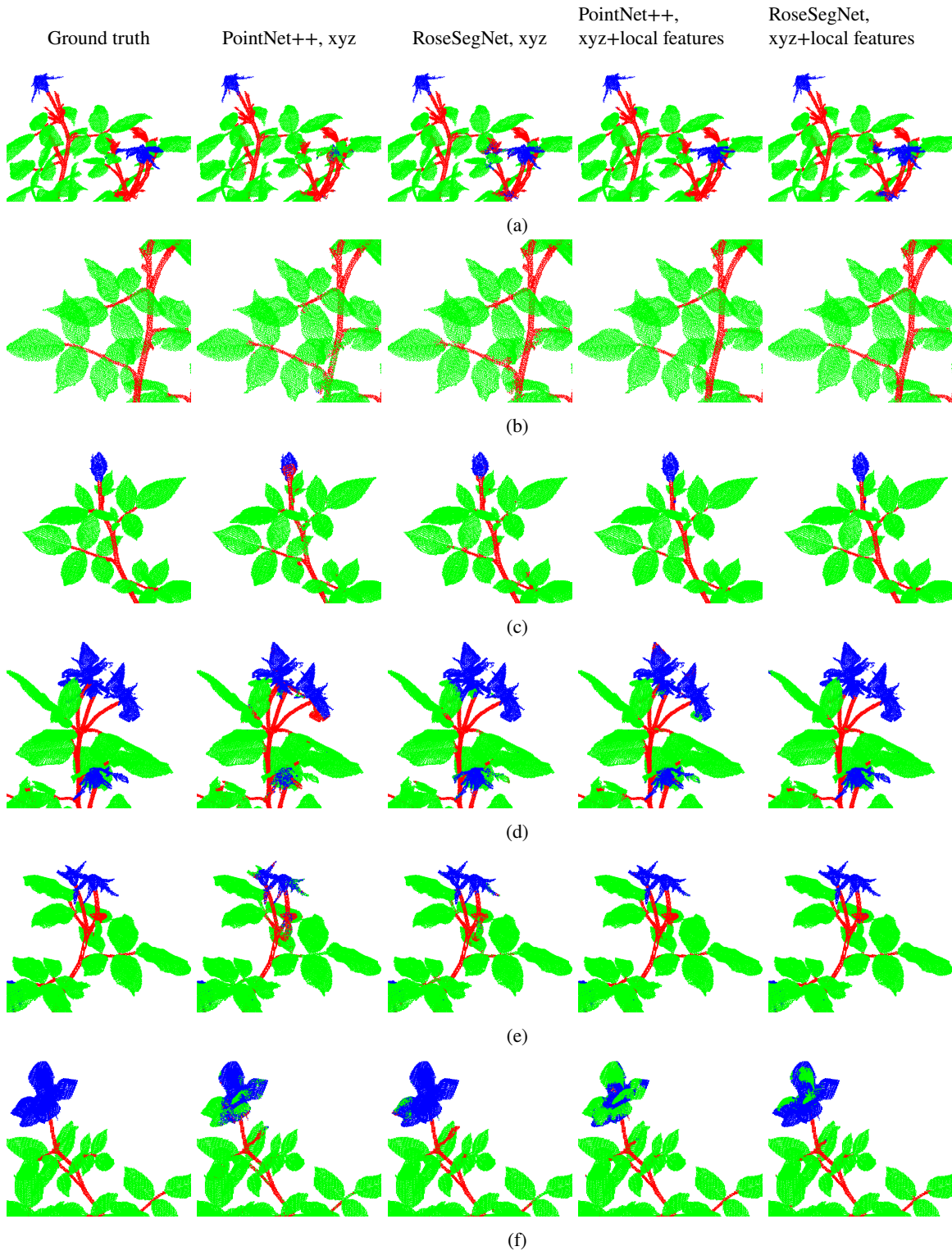


Figure 9: Segmentation results provided by PointNet++ and RoseSegNet on six sample rosebush blocks. The first column represents the ground truth. The second and third columns give the results obtained with PointNet++ and RoseSegNet, respectively, with the use of spatial coordinates only. The fourth and fifth columns depict the segmentation of the blocks by PointNet++ and RoseSegNet, respectively, when both networks are fed with local surface features.

398 The *Inter-Emb* operator provides longer range context information, which especially helps modelling the 'flower' class
399 more effectively. The number of neighbouring centre points (L) used in the *Inter-Emb* operator was also varied. Setting
400 $L = 8$ yielded the highest *MIoU* (Table 3).

401 Another experiment was conducted to assess the contribution of the residual connections. *Intra-Emb* and *Inter-Emb*
402 operators were kept, but their residual connections were removed. The *IoU* for flower and leaf classes dropped without
403 the residual connections, while there was a slight increase in *IoU* for the stem class. However, in overall, it can be
404 observed that the *MIoU* benefits from the inclusion of residual connections.

405 Finally, the number of closest points U selected for feature propagation at the *Att-Prop* module was varied. While
406 increasing this number up to a certain point ($U = 3$) had a positive effect on the performance, further increase lead to
407 a drop in performance in addition to the increased computational cost.

408 4. Discussion

409 The demand for increased productivity and quality of produce pushes for programs aiming to breed agricultural
410 plants with high genetic potential. Automated solutions for trait measurements through 3D computer vision will enable
411 experimentation with a high number of plants for breeding, genetics, and genomics research. 3D acquisition and
412 modelling of plant geometry provides complete and accurate measurements of the plant shape. Organ segmentation of
413 the 3D plant models is indispensable for extraction of organ-level traits in high-throughput phenotyping as well as for
414 monitoring emergence and growth of individual organs in horticulture.

415 Examples to such organ-level traits are number of leaves, organization of leaves, leaf areas, leaf inclination angles,
416 proportions among the organs, and shape of organs. One important application area where measurements of such traits
417 over a large number of plants during their development are in demand is genetic and mechanical control of organ
418 growth. Understanding the genetic regulation mechanisms that determine organ identity, growth, size, and shape is
419 possible through phenotypic shape measurements at organ level (Johnson & Lenhard, 2011). The role the external
420 mechanical processes and constraints play in organ morphogenesis along with genes is also an important research
421 question (Trinh et al., 2021). There is considerable variation in the shapes of plant organs such as fruits, leaves, and
422 stems even within plants with identical genetic composition. The interplay of genetics with mechanical constraints
423 leading to such variations is yet to be understood for regulating the shapes of harvestable plant organs (Lazzaro et al.,
424 2018).

425 In the case of ornamental plants, the research question of correlating the subjective aesthetic quality with
426 quantitative geometric and architectural attributes is of considerable interest (Boumaza et al., 2009; Garbez et al., 2018;
427 Demotes-Mainard et al., 2013). Garbez et al. (2018) conducted a thorough study to investigate the relation between
428 the architecture of rosebush plants and the visual perception of consumers. With a Fastrack® 3D digitiser (Polhemus,

429 Colchester, VT, USA), they manually measured a large number of traits such as apparent plant axes, their topological
430 relations (succession or branching), the number of branching, number of leaves, flowers, fruits, and carrier axes, size of
431 the leaves, height of the flowers. Demotes-Mainard et al. (2013) also explored the elements of visual quality of rosebush
432 plants by gathering quantitative attributes. They collected data of leaf dimensions via destructive measurements and
433 manually observed the number of visible leaves, internodes and terminal leaflet lengths.

434 As opposed to such laborious manual data gathering over a small number of plants, digital processing of 3D
435 plant models can provide automated trait measurements over large populations for expanding scientific knowledge
436 on the development of the shape of the produce. Also, automatic plant monitoring and shape characterization via 3D
437 vision enables accurate plant management, especially for plants where individual organ shapes as well as their spatial
438 organizations are important agronomic traits. The architectural and morphological attributes of importance, such as
439 organization of axes, leaf sizes, flower height, etc., can be estimated only through the decomposition of the acquired
440 plant model into its individual organs. The accuracy of automatic plant segmentation methods operating on 3D models
441 directly influences accurate trait estimation.

442 This work provides an organ segmentation method that brings an improvement on the accuracy of previous
443 segmentation techniques measured on a publicly available data set. The application of technical innovations in 3D
444 point cloud segmentation to plant models in the framework of deep learning is in its infancy. Deep learning techniques
445 promise fast characterization of the vast amount of structural and geometrical variations among and within plant species
446 via learning with training data, and without incorporation of much expert knowledge. The recent advances in 3D point-
447 based deep learning methods in the field of computer vision, however, mainly target robotic applications other than
448 those related to plant sciences and agriculture. The proposed RoseSegNet is a progress in the direction of designing
449 deep neural network architectures suitable for 3D plant model analysis, specifically 3D plant organ segmentation. The
450 inclusion of attention-based mechanisms modelling interactions of local structures at multiple scales, and within and
451 among local regions, is the main contribution of this work in relation with previous applications of 3D point-based
452 networks. Augmenting spatial features of local structures with hand-crafted surface features and letting the network
453 process, relate, and aggregate these features towards organ identity inference is another contribution.

454 Despite the considerable research in plant genetics, investment on provision of publicly available annotated 3D
455 plant data sets for research purposes is alarmingly low. The need for large amount of annotated data for training deep
456 learning techniques has the potential of pushing for development of fast acquisition and labelling protocols, that will
457 eventually lead to common use of 3D robot vision both in plant sciences and in agriculture. In the context of this work,
458 however, the amount of annotated data is limited to eleven rosebush models, and only one model is used for training.
459 One important research question in the framework of deep learning is the assessment of the impact of the amount of
460 training data on various architectures. The limits of the performance improvement with respect to the amount of data, as

well as the potential of substantial data availability for closing the gap between different deep learning approaches are yet to be explored. On the other hand, the ability of a network in achieving high performance with limited annotated data is of considerable importance due to the time-consuming process of manually annotating 3D models of target crops.

5. Conclusion

This paper demonstrates a progress in the improvement of organ segmentation accuracy through advanced deep learning techniques to contribute to automated and accurate estimation and monitoring of organ-level and architectural phenotypical traits that are crucial in plant sciences and horticultural processes. A novel point-based deep learning network, which is named as RoseSegNet, is proposed to segment 3D point clouds of rosebush plants into their structural parts. The network is designed to process the input point cloud in a hierarchical manner and to extract contextual features based on the relations between points. The contextual features are encoded and propagated by attention-based modules. Through ablation studies, the contribution of each of these modules was analysed. The attention-based contextual features improved the segmentation accuracy, especially for the flower class. Augmenting input point coordinates with local surface descriptors boosted the performance of RoseSegNet and PointNet++. RoseSegNet improved the segmentation performance as compared to the traditional classification methods based on local surface features. RoseSegNet also achieved significant improvement over the state-of-the-art 3D point-based deep learning framework PointNet++. These results suggest that deep learning methods devised to model 3D characteristics of plants are capable of surpassing traditional techniques that solely depend on hand-crafted features. The capacity of deep neural networks to simultaneously extract and evaluate relevant attributes from raw data and simple surface features without intervention of experts is of special importance for plant characterization. Deep neural networks trained on one plant species also have the potential of applicability to a large variety of other species through domain adaptation with the use of few training data.

Acknowledgements

The authors acknowledge the support of The Scientific and Technological Research Council of Turkey (TUBITAK), Project No: 121E088.

The authors also acknowledge the support of 2214/A International Doctoral Research Fellowship Program granted to Kaya Turgut by TUBITAK.

6. Appendix, Supplementary material

Supplementary Material: Details of the *Inter-Emb Operator*

490 **References**

- 491 Bao, Y., Tang, L., Srinivasan, S., & Schnable, P. S. (2019). Field-based architectural traits characterisation of maize plant using time-of-flight 3d
492 imaging. *Biosystems Engineering*, *178*, 86–101. doi:<https://doi.org/10.1016/j.biosystemseng.2018.11.005>.
- 493 Boogaard, F., van Henten, E., & Kootstra, G. (2021). Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds
494 with spectral data. *Biosystems Engineering*, *211*, 167–182. doi:<https://doi.org/10.1016/j.biosystemseng.2021.09.004>.
- 495 Boumaza, R., Demotes-Mainard, S., Huche-Thelie, L., & Guerin, V. (2009). Visual characterization of the esthetic quality of
496 the rosebush. *Journal of Sensory Studies*, *24*, 774–796. doi:<https://doi.org/10.1111/j.1745-459X.2009.00238.x>.
497 arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1745-459X.2009.00238.x>.
- 498 Chaudhury, A., Boudon, F., & Godin, C. (2020). 3D plant phenotyping: All you need is labelled point cloud data. In *CVPPP-ECCV 2020 -*
499 *Workshop on Computer Vision Problems in Plant Phenotyping* (pp. 1–17). Glasgow, United Kingdom. doi:[https://doi.org/10.1007/](https://doi.org/10.1007/978-3-030-65414-6_18)
500 [978-3-030-65414-6_18](https://doi.org/10.1007/978-3-030-65414-6_18).
- 501 Chaudhury, A., Hanappe, P., Boudon, R., Azais, Godin, C., & Colliaux, D. (2021). Transferring PointNet++ segmentation from virtual to real
502 plants. In *CVPPP-ICCV 2021 - 7th workshop on Computer Vision in Plant Phenotyping and Agriculture*.
- 503 Choudhury, S. D., Samal, A., & Awada, T. (2019). Leveraging image analysis for high-throughput plant phenotyping. *Frontiers in Plant Science*,
504 *10*. doi:<https://doi.org/10.3389/fpls.2019.00508>.
- 505 Demotes-Mainard, S., Bertheloot, J., Boumaza, R., HuchÃ©-ThÃ©lier, L., GuÃ©ritaine, G., GuÃ©rin, V., & Andrieu, B. (2013). Rose bush
506 leaf and internode expansion dynamics: analysis and development of a model capturing interplant variability. *Frontiers in Plant Science*, *4*.
507 doi:[10.3389/fpls.2013.00418](https://doi.org/10.3389/fpls.2013.00418).
- 508 Dey, D., Mummert, L., & Sukthankar, R. (2012). Classification of plant structures from uncalibrated image sequences. In *2012 IEEE Workshop on*
509 *the Applications of Computer Vision (WACV)* (pp. 329–336). doi:<https://doi.org/10.1109/WACV.2012.6163017>.
- 510 Dutagaci, H., Rasti, P., Galopin, G., & Rousseau, D. (2020). ROSE-X: An annotated data set for evaluation of 3D plant organ segmentation methods.
511 *Plant Methods*, *16*. doi:<https://doi.org/10.1186/s13007-020-00573-w>.
- 512 Elnashef, B., Filin, S., & Lati, R. N. (2019). Tensor-based classification and segmentation of three-dimensional point clouds for organ-level plant
513 phenotyping and growth analysis. *Computers and Electronics in Agriculture*, *156*, 51–61. doi:[https://doi.org/10.1016/j.compag.2018.](https://doi.org/10.1016/j.compag.2018.10.036)
514 [10.036](https://doi.org/10.1016/j.compag.2018.10.036).
- 515 Feldmann, M. J., & Tabb, A. (2022). Cost-effective, high-throughput phenotyping system for 3D reconstruction of fruit form. *The Plant Phenome*
516 *Journal*, *5*. doi:<https://doi.org/10.1002/ppj2.20029>.
- 517 Garbez, M., Symoneaux, R., Belin, E., Caraglio, y., Chéné, y., Dones, N., Durand, J.-B., Hunault, G., Relion, D., Sigogne, M., Rousseau, D., &
518 Galopin, G. (2018). Ornamental plants architectural characteristics in relation to visual sensory attributes: a new approach on the rose bush for
519 objective evaluation of the visual quality. *European Journal of Horticultural Science*, *83*, 187–201. doi:[10.17660/eJHS.2018/83.3.8](https://doi.org/10.17660/eJHS.2018/83.3.8).
- 520 Ghahremani, M., Williams, K., Corke, F., Tiddeman, B. P., Liu, Y., & Doonan, J. H. (2021). Deep segmentation of point clouds of wheat. *Frontiers*
521 *in Plant Science*, *12*. doi:<https://doi.org/10.3389/fpls.2021.608732>.
- 522 Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., & Hu, S.-M. (2021a). PCT: Point cloud transformer. *Computational Visual Media*, *7*,
523 187–199. doi:<https://doi.org/10.1007/s41095-021-0229-5>.
- 524 Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021b). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern*
525 *Analysis and Machine Intelligence*, *43*, 4338–4364. doi:<https://doi.org/10.1109/TPAMI.2020.3005434>.
- 526 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and*
527 *Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (pp. 770–778). doi:<https://doi.org/10.1109/CVPR.2016.90>.

- 528 Japes, B., Mack, J., Rist, F., Herzog, K., Töpfer, R., & Steinhage, V. (2018). Multi-view semantic labeling of 3D point clouds for automated plant
529 phenotyping. *ArXiv, abs/1805.03994*.
- 530 Jin, S., Su, Y., Gao, S., Wu, F., Hu, T., Liu, J., Li, W., Wang, D., Chen, S., Jiang, Y., Pang, S., & Guo, Q. (2018). Deep learning: Individual
531 maize segmentation from terrestrial lidar data using faster R-CNN and regional growth algorithms. *Frontiers in Plant Science, 9*, 866.
532 doi:<https://doi.org/10.3389/fpls.2018.00866>.
- 533 Jin, S. et al. (2020). Separating the structural components of maize for field phenotyping using terrestrial lidar data and deep convolutional neural
534 networks. *IEEE Transactions on Geoscience and Remote Sensing, 58*, 2644–2658. doi:<https://doi.org/10.1109/TGRS.2019.2953092>.
- 535 Johnson, K., & Lenhard, M. (2011). Genetic control of plant organ growth. *New Phytologist, 191*, 319–333. doi:<https://doi.org/10.1111/j.1469-8137.2011.03737.x>.
536 arXiv:<https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.2011.03737.x>.
- 537 Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2021). Transformers in vision: A survey. doi:<https://doi.org/10.1145/3505244>.
538
- 539 Lazzaro, M. D., Wu, S., Snouffer, A., Wang, Y., & van der Knaap, E. (2018). Plant organ shapes are regulated by protein interactions and associations
540 with microtubules. *Frontiers in Plant Science, 9*. doi:10.3389/fpls.2018.01766.
- 541 Le Louëdec, J., & Cielniak, G. (2021). 3D shape sensing and deep learning-based segmentation of strawberries. *Computers and Electronics in
542 Agriculture, 190*, 106374. doi:<https://doi.org/10.1016/j.compag.2021.106374>.
- 543 Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). PointCNN: Convolution on $\{X\}$ -transformed points. In *Advances in Neural Information
544 Processing Systems*. Curran Associates, Inc. volume 31.
- 545 Liu, X., Hu, C., & Li, P. (2020a). Automatic segmentation of overlapped poplar seedling leaves combining mask R-CNN and DBSCAN. *Computers
546 and Electronics in Agriculture, 178*, 105753. doi:<https://doi.org/10.1016/j.compag.2020.105753>.
- 547 Liu, Z., Zhang, Q., Wang, P., Li, Z., & Wang, H. (2020b). Automated classification of stems and leaves of potted plants based on point cloud data.
548 *Biosystems Engineering, 200*, 215–230. doi:<https://doi.org/10.1016/j.biosystemseng.2020.10.006>.
- 549 Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., & Whiting, M. D. (2020). Deep learning based segmentation for automated training
550 of apple trees on trellis wires. *Computers and Electronics in Agriculture, 170*, 105277. doi:<https://doi.org/10.1016/j.compag.2020.105277>.
551
- 552 Minervini, M., Schar, H., & Tsafaris, S. A. (2015). Image analysis: The new bottleneck in plant phenotyping [applications corner]. *IEEE Signal
553 Processing Magazine, 32*, 126–131. doi:<https://doi.org/10.1109/MSP.2015.2405111>.
- 554 Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., & Melgani, F. (2018). Computer vision-based phenotyp-
555 ing for improvement of plant productivity: a machine learning perspective. *GigaScience, 8*. doi:10.1093/gigascience/giy153.
556 arXiv:<https://academic.oup.com/gigascience/article-pdf/8/1/giy153/27359533/giy153.pdf>. Giy153.
- 557 Morel, J., Bac, A., & Kanai, T. (2020). Segmentation of unbalanced and in-homogeneous point clouds and its application to 3D scanned trees. *The
558 Visual Computer (Special issue for CGI 2020), 36*, 2419–2431.
- 559 Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE
560 Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 77–85). doi:<https://doi.org/10.1109/CVPR.2017.16>.
- 561 Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS'17:
562 Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- 563 Rosu, R. A., Schütt, P., Quenzel, J., & Behnke, S. (2020). Latticenet: Fast point cloud segmentation using permutohedral lattices. In *Proc. of
564 Robotics: Science and Systems (RSS)*.

- 565 Schunck, D., Magistri, F., Rosu, R. A., Cornelißen, A., Chebroly, N., Paulus, S., Léon, J., Behnke, S., Stachniss, C., Kuhlmann, H., & Klingbeil, L.
566 (2021). Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. *PLoS ONE*,
567 *16*, e0256340. doi:<https://doi.org/10.1371/journal.pone.0256340>.
- 568 Shi, W., van de Zedde, R., Jiang, H., & Kootstra, G. (2019). Plant-part segmentation using deep learning and multi-view vision. *Biosystems*
569 *Engineering*, . doi:<https://doi.org/10.1016/j.biosystemseng.2019.08.014>.
- 570 Sodhi, P., Vijayarangan, S., & Wettergreen, D. (2017). In-field segmentation and identification of plant structures using 3D imaging. In *2017*
571 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5180–5187). doi:[https://doi.org/10.1109/IROS.](https://doi.org/10.1109/IROS.2017.8206407)
572 *2017.8206407*.
- 573 Trinh, D.-C., Alonso-Serra, J., Asaoka, M., Colin, L., Cortes, M., Malivert, A., Takatani, S., Zhao, F., Traas, J., Trehin, C., & Hamant, O. (2021).
574 How mechanical forces shape plant organs. *Current Biology*, *31*, R143–R159. doi:<https://doi.org/10.1016/j.cub.2020.12.001>.
- 575 Turgut, K., Dutagaci, H., Galopin, G., & Rousseau, D. (2022). Segmentation of structural parts of rosebush plants with 3D point-based deep learning
576 methods. *Plant Methods*, *18*. doi:<https://doi.org/10.1186/s13007-022-00857-3>.
- 577 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In
578 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing*
579 *Systems*. volume 30.
- 580 Wahabzada, M., Paulus, S., Kersting, K., & Mahlein, A.-K. (2015). Automated interpretation of 3D laserscanned point clouds for plant organ
581 segmentation. *BMC Bioinformatics*, *16*. doi:<https://doi.org/10.1186/s12859-015-0665-2>.
- 582 Wang, J., Chen, X., Cao, L., An, F., Chen, B., Xue, L., & Yun, T. (2019). Individual rubber tree segmentation based on ground-based lidar data and
583 faster R-CNN of deep learning. *Forests*, *10*. doi:<https://doi.org/10.3390/f10090793>.
- 584 Wang, X., Jin, Y., Cen, Y., Wang, T., & Li, Y. (2021). Attention models for point clouds in deep learning: A survey. *ArXiv, abs/2102.10788*.
- 585 Xu, L., He, K., & Lv, J. (2019). Bayberry image segmentation based on manifold ranking salient object detection method. *Biosystems Engineering*,
586 *178*, 264–274. doi:<https://doi.org/10.1016/j.biosystemseng.2018.12.001>.
- 587 Zhang, L., Xia, C., Xiao, D., Weckler, P., Lan, Y., & Lee, J. M. (2021). A coarse-to-fine leaf detection approach based on leaf skeleton identification
588 and joint segmentation. *Biosystems Engineering*, *206*, 94–108. doi:<https://doi.org/10.1016/j.biosystemseng.2021.03.017>.
- 589 Zhao, H., Jiang, L., Jia, J., Torr, P. H., & Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF International Conference on*
590 *Computer Vision* (pp. 16259–16268).
- 591 Ziamtsov, I., & Navlakha, S. (2019). Machine learning approaches to improve three basic plant phenotyping tasks using three-dimensional point
592 clouds. *Plant Physiology*, *181*, 1425–1440. doi:<https://doi.org/10.1104/pp.19.00524>.