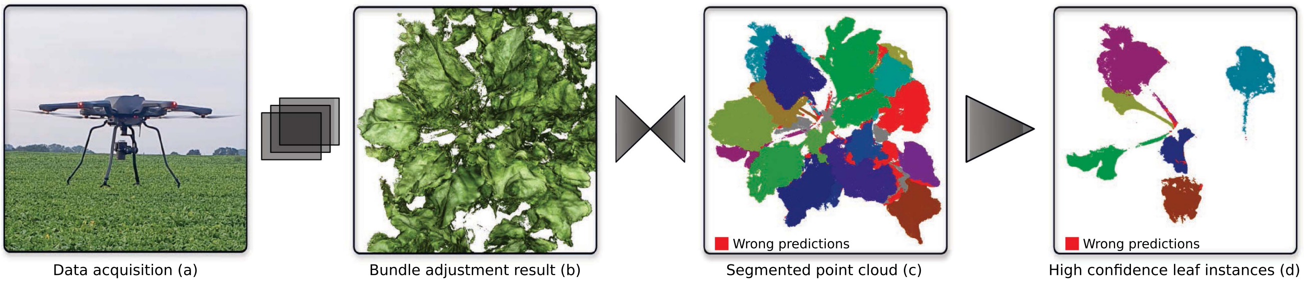
High Precision Leaf Instance Segmentation for Phenotyping in Point Clouds Obtained Under Real Field Conditions

Elias Marks, *Student Member, IEEE*, Matteo Sodano, Federico Magistri, *Graduate Student Member, IEEE*, Louis Wiesmann, *Graduate Student Member, IEEE*, Dhagash Desai, Rodrigo Marcuzzi, *Graduate Student Member, IEEE*, Jens Behley, *Member, IEEE*, and Cyrill Stachniss, *Member, IEEE*



Data acquisition (a)

Bundle adjustment result (b)

Segmented point cloud (c)

High confidence leaf instances (d)

Fig. 1. Workflow to obtain high precision segmentations of crop leaves. We collect images on the field (a), process them into point clouds (b) by using bundle adjustment. We feed these into our segmentation network to obtain leaf instance candidates (c). As done in current breeding practices, we use a subset of leaves for phenotypic trait evaluation. We select these leaves with the highest confidence based on a predicted confidence score to alleviate the influence of the segmentation errors on the phenotyping process (d).

***Abstract*—Measuring plant traits with high throughput allows breeders to monitor and select the best cultivars for subsequent breeding generations. This can enable farmers to improve yield to produce more food, feed, and fiber. Current breeding practices involve extracting leaf parameters on a small subset of the leaves present in the breeding plots, while still requiring substantial manual labor. To automate this process, an important step is the precise distinction between separate leaves, which is the problem we address in this letter. We exploit recent advancements in 3D deep learning to build a convolutional neural network that learns to segment individual leaves. As done in current breeding practices, we select a subset of leaves to be used for phenotypic trait evaluation as this allows us to alleviate the influence of segmentation errors on the phenotypic trait estimation. To this extent we propose to use an additional neural network to predict the quality of each segmented leaf and discard inaccurate leaf instances. The experiments show that our network yields higher segmentation accuracy on sugar beet breeding plots planted under the supervision of the German Federal Office for Plant Varieties. Furthermore, we show that our neural network helps in filtering out leaves with lower segmentation accuracy.**

***Index Terms*—Agricultural automation, robotics and automation in agriculture and forestry, deep learning for visual perception.**

I. Introduction

PLANT phenotyping is a crucial tool for plant breeding and crop production. It analyses the visible traits of plants to understand the physical and physiological plant development [8]. Breeders measure individual plants and their organs, i.e., leaves, and fruits, regularly to select the best cultivars to be used for the following breeding generations. This process is fundamental for providing food, feed, and fiber for the growing world population as new species are targeted to increase the productivity and adaptability of crops. However, common breeding practices involve substantial human labor to estimate plant features, even when evaluating the traits only on a small subset of the crop canopies. This is leading to measurements with low throughput and low repeatability, especially for the

Manuscript received 4 April 2023; accepted 31 May 2023. Date of publication 21 June 2023; date of current version 3 July 2023. This letter was recommended for publication by Associate Editor A. Degani and Editor H. Moon upon evaluation of the reviewers' comments. This work was supported in part by Deutsche Forschungsgemeinschaft (DFG), German Research Foundation through Germany's Excellence Strategy, under Grant EXC-2070 – 390732324 – PhenoRob and in part by the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) through the Innovation Support Programme under Grant 28DK108B20 (RegisTer).

Elias Marks, Matteo Sodano, Federico Magistri, Louis Wiesmann, Dhagash Desai, Rodrigo Marcuzzi, and Jens Behley are with the University of Bonn, 53115 Bonn, Germany (e-mail: elias.marks@uni-bonn.de; matteo.sodano7@gmail.com; federico.magistri@igg.uni-bonn.de; louis.wiesmann@igg.uni-bonn.de; desaidhagash@gmail.com; rodrigomarcuzzi@gmail.com; jens.behley@igg.uni-bonn.de).

Cyrill Stachniss is with the University of Bonn, 53115 Bonn, Germany, also with the Department of Engineering Science, University of Oxford, Oxford, U.K., and also with the Lamarr Institute for Machine Learning and Artificial Intelligence, 44227 Dortmund, Germany (e-mail: cyrill.stachniss@igg.uni-bonn.de).

Digital Object Identifier 10.1109/LRA.2023.3288383

more subjective parameters dependent on color and lighting [6], [9].

For high-throughput and cost-efficient objective measurements, a promising approach is the usage of unmanned aerial vehicles (UAVs) for collecting data and deep learning pipelines to analyze the data. The main advantage of collecting data using UAVs is that they can cover large breeding plots in a small time frame using high-resolution cameras.

In most applications UAVs acquire images from a single viewpoint (top-down perspective) and only analyze single images, i.e., 2D data. This approach, however, makes it difficult to extract three-dimensional traits in breeding plots, which are densely covered with plants, leading to many parts of the crops being occluded from the UAV perspective. We alleviate this issue by collecting UAV image data from different viewpoints, leading to better coverage the lower parts of the crops. We then use these images to compute high-resolution point clouds by using bundle adjustment.

We address the task of detecting and segmenting leaves in densely planted agricultural plots, using the aforementioned point clouds generated out of high-resolution cameras mounted on a UAV. The task of leaf segmentation is a highly relevant step for autonomous phenotyping and the quality of the segmented leaves is of high importance, as the variations in the traits that need to be measured by the breeders are often very small. In current breeding practices, only a subset of the leaves present in the breeding plots are actually used for phenotypic trait estimation, as breeders are interested in the average trait per plot and a subset of leaves is sufficient to compute this [6]. We also select a subset of leaves to mitigate segmentation errors, which still affect current state of the art 3D deep learning approaches, and otherwise would negatively impact the phenotypic trait extraction, see Fig. 1. We target the monitoring of sugar beet plants in commercial breeding plots grown and maintained under the supervision of the German Federal Office of Plant Varieties (Bundessortenamt). In the agricultural context, the task of segmenting leaf canopies into single leaves from noisy and partial 3D point cloud data has not been tackled in a unified manner, given the cluttered nature of the scene and the lack of publicly available datasets to benchmark new approaches. However, being able to segment individual leaves allows for precise 3D reconstruction and estimation of important phenotypic traits [15], [16], [18].

The main contribution of this letter is a novel deep learning method to segment crop leaves in field point clouds with high precision. We perform this operation on patches that are extracted automatically to achieve full coverage of each breeding plot, minimizing the amount of manual labor involved. As done in current breeding practices, we select a subset of the leaves for subsequent trait extraction. We do this to alleviate the influence of the segmentation errors on the phenotyping process, since the accuracy of the extracted traits is very important to detect even minor differences between varieties. We, therefore, aim for discarding inaccurate leaf predictions, which allows us to minimize segmentation errors while still evaluating substantially more leaves than required, i.e., by the phenotyping guidelines of the European Commission [6]. To achieve this, we predict leaf instance masks and an associated confidence score with a deep neural network. Based on the estimated confidences, we then filter out the predictions with the lowest confidence scores, which allows us to keep only the more accurate leaf masks.

In sum, we make three key claims: Our approach is able to (i) improve leaf segmentation performance on manually segmented plant point clouds; (ii) robustly segment leaves from point cloud patches of real breeding fields with high measurement noise; (iii) filter out inaccurate leaf masks based on a confidence prediction to increase the quality of the leaf segmentations. These claims are backed up by the paper and our experimental evaluation.[1]

II. Related Work

Multiple works have been proposed for leaf segmentation [10], [21], [22], [30], [31]. Most of these however work on 2D image data as this is readily available and methods for image segmentation have been developed early on [4], [11]. Weyler et al. [30], [31] propose a deep learning approach to predict offset vectors pointing to the center of the leaves and plants. They then cluster the pixels into individual leaves and plants based on the predicted offsets. The approach by Guo et al. [10] instead directly predicts the leaf masks, skipping the clustering post-processing step. Roggiolani et al. [21], [22] propose an approach to segment individual plants and their leaves while exploiting the hierarchical structure of the task. Our work instead works on point clouds as we are interested in the 3D structure of the plants.

Recently, a diverse number of works proposed deep learning methods to process 3D data, either by using voxel grids [5], rendering multi-view images [27], employing point operators [20], or by defining convolutions on point clouds [28]. Similar to images, some approaches base their predictions on estimating offset vectors and subsequently clustering of points into single instances [13], [29]. Instead, to achieve end-to-end segmentation without a post-processing step the mask-based approaches show promising results on point clouds [17], [24].

Above-mentioned works address the segmentation of autonomous driving scenarios and indoor environments, our work instead focuses on agricultural field data, which present different challenges, especially the deformability of the scene causing high levels of noise induced by meteorologic factors. One of the major bottlenecks for point cloud segmentation in the agricultural setting is the lack of publicly available datasets for instance segmentation. To tackle this issue, Roggiolani et al. [21], [23] propose a self-supervised pretraining mechanism to better initialize the neural network, thus needing less training data. To the best of our knowledge, Schunk et al. [25] released the only publicly available dataset with leaf instances tracked over time. In this dataset, however, the plants are scanned with a high-precision laser scanner in laboratory conditions, a data acquisition approach that is not feasible for our application. Using similar data, Heiwolt et al. [12] segment plants into individual organs using a deep architecture operating on points. Shi et al. [26] instead use a multi-view neural network predicting instances on images and then aggregating them into point clouds.

In contrast to the aforementioned approaches, our approach works directly on point clouds acquired using UAV imagery in densely planted agricultural plots containing a multitude of different varieties leading to a big variation in the leaf shapes. Also, the data acquisition was performed under the influence of typical field conditions such as wind, heat, and variation of the light conditions. This leads to high levels of noise in the point clouds as exemplified in Fig. 3.

[1] The open source implementation is [Online]. Available: https://github.com/PRBonn/plant_pcd_segmenter

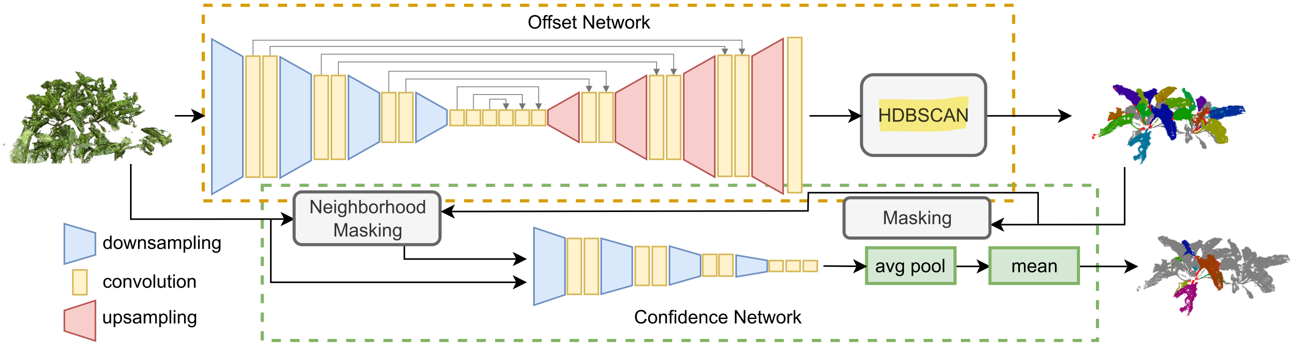


Fig. 2. The architecture of our leaf segmentation approach. The layers in the yellow box are part of our autoencoder network for the prediction of the offset vectors. In the green box we show the encoder network for predicting the confidence of each instance.

III. Our Approach

We propose a deep learning approach designed for detecting individual leaves of real crop plants in 3D point cloud data acquired in breeding plots, see Fig. 2. We aim to segment each point cloud $\mathcal{S}$ containing a portion of a breeding plot, from now on referred to as field patch, into disjoint subsets $\mathcal{S}_i$, representing the individual leaves.

To perform the segmentation, we predict a spatial offset vector for each point from its location to the center of the corresponding leaf. We then shift the points by the offsets and cluster such leaf center estimates into separate instances.

The targeted use for our leaf segmentations is extracting phenotypic traits, which often show only minor differences between varieties. Therefore, our main focus is having highly precise segmentation results. As in current breeding practices, we select a subset of leaves for evaluation and do so to mitigate false positives. To this extent, we propose a second network to predict the accuracy of each leaf prediction which allows us to filter out the most inaccurate predictions.

A. Leaf Instance Segmentation

To cluster the points into individual leaves, we leverage a deep neural network to predict an offset vector for each point, pointing towards the leaf center. The encoder part of this offset prediction network takes a field patch point cloud $\mathcal{S}$ as input. Those point clouds are generated from a high-resolution camera mounted on a UAV using bundle adjustment. To provide additional information to the network, we input also point positions and colors as input features for the first encoding block. Our encoder is composed of encoding blocks E_i that use KPConv [28] as the backbone, which is a form of convolution working directly on 3D points in Euclidean space. For more details, we refer to the original publication by Thomas et al. [28].

Each encoding block E_i contains a pre-activation and post-activation block, composed of a linear layer, layer norm, and Leaky ReLU activation. Each of these blocks E_i takes as input the features F_{i-1}^E from the previous block E_{i-1} and outputs the features F_i^E.

To achieve an increase of the feature scale, we successively increase the kernel radius r of the convolution after each downsampling operation to get

$$r_i = r_{\min} + r_{step}\, i, \tag{1}$$

where $r_{\min}$ is the user defined smallest kernel radius, $r_{step} = (r_{\max} - r_{\min})/n_d$, $r_{\max}$ is the maximum kernel radius defined by the user, and n_d is the number of downsampling steps. To save compute time, we perform downsampling on the point cloud after each increase of the kernel size r. To perform this downsampling operation, we divide the Euclidean space $\mathbb{R}^3$ into voxels. For each voxel, we compute the centroid of the subset of points falling within it and output these centroids as the resulting point cloud. The voxel size of the grid sampling algorithm is set as $d_s = r_i/\sigma$, where σ is a user-defined ratio.

We also increase the feature depth s_o^i at each downsampling step from $s_o^{\min}$ to $s_o^{\max}$ to allow the network to learn more complex features at the deeper levels. This increase is performed at equal steps in the same manner as for the kernel radius r, see (1). In sum, our encoder architecture is composed of $n_e = 9$ encoding blocks and $n_d = 4$ downsampling steps. The actual sequence of these operations is shown in Fig. 2.

In the decoder, we use a sequence of upsampling blocks followed by a multi-layer perceptron (MLP) to predict the offset vectors. The upsampling blocks are again composed of KPConv blocks. To support the decoding phase with high-resolution details, the output features F_i^D of each decoding block are are then summed to the features F_i^E originating from the encoding block E_i at the same stage. The MLP is composed of 4 linear layers combined with Leaky ReLU activations, has $s_o^{\max}$ input channels and 3 output channels. The size of the layers in the MLP decreases in a stepwise manner from $s_o^{\max}$ to the size 3 of the spatial offset vectors. By applying the network on every point, we end up with the set of offset vectors $\mathcal{V}_i$, containing a predicted offset $\boldsymbol{v}_j$ for every point $\boldsymbol{p}_j$ in the input point cloud $\mathcal{S}$.

To group the points into separate leaves we use the bottom-up hierarchical clustering method HDBSCAN [19], by using the distance function

$$\gamma = \|(\boldsymbol{p}_j + \boldsymbol{v}_j) - (\boldsymbol{p}_k + \boldsymbol{v}_k)\|_2. \tag{2}$$

Given a set of points, the algorithm finds core points of high density and expands clusters from them, while it marks as outliers points that are in low-density regions. The algorithm performs DBSCAN over varying density values ϵ and integrates the result to find a clustering with the most stable result when changing ϵ. This improves the algorithms robustness to variations in the densities. The output of this postprocessing step are the individual leaf instances $\mathcal{S}_i$.

B. Confidence Estimation of the Leaf Instances

We discard the inaccurate leaf instance predictions to minimize the influence of the segmentation errors on the phenotypic feature extraction. Therefore, we design a second network to estimate the accuracy of each leaf instance by assigning a confidence score to them. We compute the confidence of each instance $\mathcal{S}_i$ separately, as we found this to work better than estimating them all at once. The confidence network is composed of an encoder, which has the same topology as the one used in the offset network.

We feed the network with the points $\mathcal{P}_n := \{\boldsymbol{p} \mid d_{\min}(\boldsymbol{p}, \mathcal{S}_i) < d_{lim}\}$ with $\boldsymbol{p} \in \mathcal{S}$, where $d_{\min}$ is the function that returns the minimum Euclidean distance to the points contained in the leaf instance prediction $\mathcal{S}_i$ and d_{lim} is a selectable threshold. The point cloud $\mathcal{P}_n$ is therefore the set of all points $\boldsymbol{p}$ in the neighborhood of the leaf prediction $\mathcal{S}_i$. Together with $\mathcal{P}_n$, we also input a set of feature vectors $\mathcal{F}_c$ into the network. To obtain these features $\mathcal{F}_c$, we compute the 3D vector $\boldsymbol{\delta}_j$ containing the difference between the center prediction $\boldsymbol{c}^i_{pred} = \boldsymbol{p}_j + \boldsymbol{v}_j$ associated to the point and the mean center prediction of the leaf

$$\boldsymbol{\delta}_j = \boldsymbol{c}^i_{pred} - \sum_j \frac{\boldsymbol{c}^i_{pred}}{|\mathcal{V}_i|}, \tag{3}$$

where $\mathcal{V}_i$ is the set of offset predictions of the points $\mathcal{S}_i$ predicted as part of a leaf instance i. Then, we define $\mathcal{F}_c = \{\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{|\mathcal{V}_i|}\}$ with

$$\boldsymbol{f}_j = \begin{cases} \boldsymbol{\delta}_j, & \text{if } \boldsymbol{p}_j \in \mathcal{S}_i \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

By providing the input features $\mathcal{F}_c$ instead of the instance masks $\boldsymbol{m}_i$, defined as:

$$\boldsymbol{m}_i = \begin{cases} 1, & \text{if } \boldsymbol{p}_j \in \mathcal{S}_i \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

to the network, one provides information about the distribution of the leaf center predictions. This information intuitively can help to better estimate the confidence value, compared to providing only which points belong to the leaf instance.

The output of the last convolution block is a feature vector of defined size $s_c^{\max}$ for each point produced by the last downsampling step. These points are then fed into an MLP, which outputs for each point a feature vector $\boldsymbol{f}_c^j$ of size $s_c^{\max}$.

We perform voxel downsampling operations on the instance mask $\boldsymbol{m}_i$ to obtain the downsampled version $\boldsymbol{m}_i^{out}$ defining the predicted instance in the downsampled point cloud. To get the IoU prediction for the current instance, we then apply average pooling to all points defined by $\boldsymbol{m}_i^{out}$. We then obtain the IoU estimate for the current instance by average pooling the output vectors and taking the mean of all entries. To leverage the confidence prediction of each leaf we fix a ratio $r_{keep} = n_{keep}/n_{tot}$, where n_{keep} is the number of kept leaves and n_{tot} is the total number of leaves. We then compute the minimum confidence value $c_{\min}$ in order to obtain the percentile given by r_{keep} and discard all leaf instance with a predicted confidence below that threshold.

C. Loss Functions

To supervise the network to predict offset vectors for accurate leaf instance segmentation, we first compute the leaf center $\boldsymbol{c}_i$ for each leaf point cloud $\mathcal{S}_i$ as follows:

$$\boldsymbol{c}_i = \frac{1}{|\mathcal{S}_i|} \sum_{\boldsymbol{p} \in \mathcal{S}_i} \boldsymbol{p}. \tag{6}$$

Then, we compute the offset labels $\boldsymbol{v}^*_j = \boldsymbol{p}_j - \boldsymbol{c}_i, \mid \boldsymbol{p}_j \in \mathcal{S}_i$ used for the supervision. $\mathcal{V}^* := \{\boldsymbol{v}^*_j = \boldsymbol{p}_j - \boldsymbol{c}_i, \mid \boldsymbol{p}_j \in \mathcal{S}_i, \, \mathcal{S}_i \in \mathcal{S}\}$ We then use an L1 loss for the optimization of the network weights with $\boldsymbol{v}_j \in \mathcal{V}$ and $\boldsymbol{v}^*_j \in \mathcal{V}^*$:

$$\mathcal{L}_{offset} = \frac{\sum_i |\boldsymbol{v}_j - \boldsymbol{v}^*_j|}{|\mathcal{S}|}. \tag{7}$$

To compute the loss $\mathcal{L}_{confid}$ of the IoU prediction network, we need reference values IoU^*_k for each predicted leaf instance. To obtain these labels we compute the intersection of the predicted instance mask $\boldsymbol{m}_i$ with all ground truth instance labels $\boldsymbol{m}^*_i$:

$$\text{IoU}_k = \frac{|\boldsymbol{m}_i \cap \boldsymbol{m}^*_i|}{|\boldsymbol{m}_i \cup \boldsymbol{m}^*_i|}, \, \boldsymbol{m}^*_i \in \text{GT}, \tag{8}$$

where GT is the set of ground truth leaf instances. Then we take the highest intersection as our reference value:

$$\mathcal{O}_i = \{\text{IoU}_0, \ldots, \text{IoU}_{n_{gt}}\} \tag{9}$$

$$\text{IoU}^*_i = \max(\mathcal{O}_i), \tag{10}$$

where $\mathcal{O}_i$ is the set of all IoU_k and $n_{gt} = |\text{GT}|$ is the number of ground truth leaf instances. We then compute the L1 loss between the predicted IoU values IoU_i and the corresponding IoU^*_i. By discarding leaf predictions in the filtering procedure based on the predicted IoU values IoU_i, we want to make sure that all kept instances are high-accuracy predictions. Therefore, we favor underestimations instead of overestimations by the confidence network, as overestimations often lead to remaining predictions with low accuracy after filtering. To induce this behavior, we increase the weight of the overestimated predictions in the loss computation, leading to the loss being computed as

$$\mathcal{L}_{confid} = \sum_j a_i \, \frac{|\boldsymbol{p}_j - \boldsymbol{v}^*_j|}{|\mathcal{S}|} \tag{11}$$

with

$$a_i = \begin{cases} \omega, \text{if } \text{IoU}_i > \text{IoU}^*_i \\ 1, \text{ otherwise,} \end{cases} \tag{12}$$

where ω is a user-defined weight.

IV. Experimental Evaluation

The main focus of this work is an approach that accurately segments leaf instances in the point cloud of a field, while discarding low accuracy instances to alleviate the influence of segmentation errors on the phenotypic trait extraction.

We present our experiments to analyze the capabilities of our method. The results of our experiments support our key claims, which are: (i) improve leaf segmentation performance on manually segmented plant point clouds; (ii) robustly segment leaves from point cloud patches of real breeding fields with high measurement noise; (iii) filter out inaccurate leaf masks based on a confidence prediction to increase the quality of the leaf segmentations.

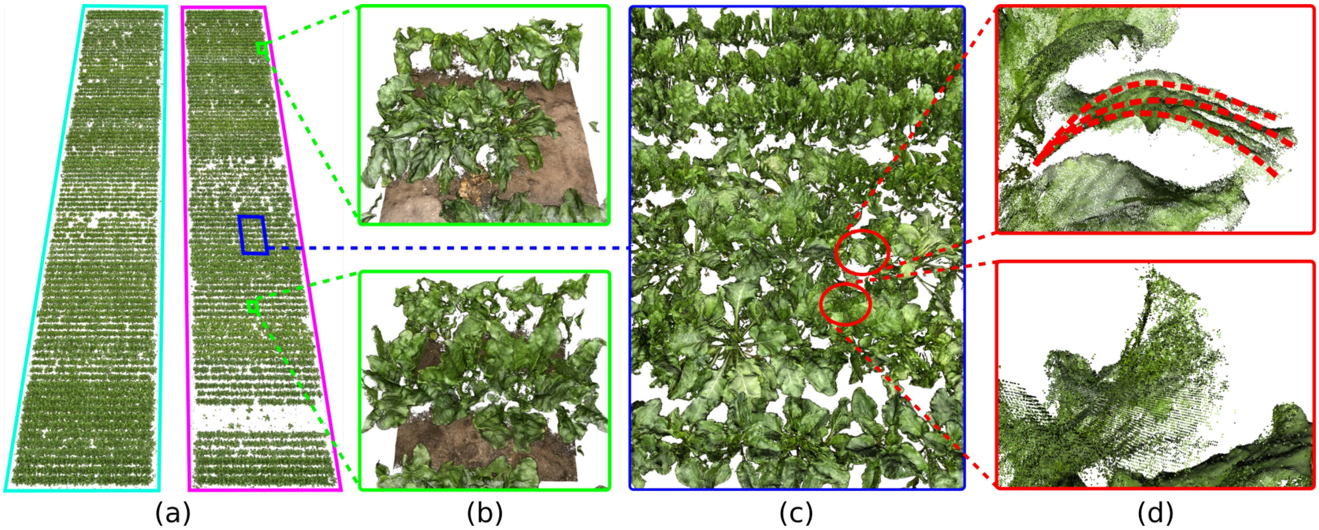


Fig. 3. Data used by our approach. (a) Shows the 3D point cloud of the field using bundle adjustment, with the train and test data collected in the pink row and the test data in the cyan row. (b) Shows the field patches as they are used as input for the network. We also show how crowded the breeding plots are (c). In the top image of (d) we show noise induced by deformation due to high temperature which leads to the same leaf showing up multiple times in the data and in the bottom noise due to wind which can be seen as scattered points.

A. *Datasets*

We trained and evaluated our approach on a dataset composed of sugar beet plant point clouds. We generated the data by imaging breeding plots containing multiple varieties of the crops, to ensure better generalization capabilities of the trained models, and processing the resulting 100-megapixel images into 3D point clouds using bundle adjustment. As sensor, we used the PhaseOne iXM-100 camera attached to a UAV. We flew three missions over the breeding field at 21 m height from ground and with a camera angle of 45°, 90°, and 135° from the ground plane as this leads to the best photogrammetric reconstruction. This approach leads to a 10-fold increase in the computation time but enables coverage of the full crops including the lower parts even in advanced growth stages. To ensure that the test data remains unseen during the training phase we defined a test region in the field that is used exclusively for evaluation purposes.

The captured breeding field contains a large number of varieties and a huge amount of leaves, making the labeling of all crops intractable. We, therefore, manually annotated the leaf IDs of small groups of adjacent plants. The annotation process was carried out by different people to avoid human biases. The resulting labels enable the networks to learn the interconnections between plants while also covering the varieties on the field.

As the point cloud of the whole field contains 11.6 billion points, it is impossible to process it at once. Thus, we extracted patches of 1×1 m overlapping by 50% in both directions. This resulted in 37 patches for training, 10 for validation, and 69 for testing. We show the field, an overview of the data, and examples of noise in the dataset in Fig. 3.

B. *Training Procedure*

We train our networks on an Nvidia Quadro RTX A6000 with 24 GB of memory. We generate input batches of 10 point clouds containing 100,000 points by randomly subsampling the original point clouds. We use elastic deformation [3], random rotation, and axis flipping as augmentations to improve generalization with the relatively small training set.

We use a learning rate of 10^{-4}, nine encoder and decoder blocks, four downsampling and four upsampling operations, and a maximum feature size $s_o^{\max} = 256$. We generally found that with a smaller maximum convolution kernel size $r_{\max}$ the performance increases but so does the memory usage and compute time. We use $r_{\min} = 0.006$ and $r_{\max} = 0.08$, which allows us to train with a batch size of 4. We trained the segmentation network for 500 epochs.

For the confidence estimation task, a smaller number of parameters is sufficient, which leads us to use a maximum feature size $s_c^{\max}$ of 96 and nine convolutional encoder and decoder blocks. The lower amount of input points and network parameters enabled us to use a smaller $r_{\max} = 0.02$ and the same $r_{\min} = 0.006$ while still allowing us to train with a batch size of 10. We set the weight ω in the confidence loss $\mathcal{L}_{confid}$ to 2. We train the confidence network for 200 epochs. After convergence, we evaluate the performance of our model in the following experiments.

C. *Leaf Instance Segmentation on Field Patches*

The first experiment evaluates the performance of our approach in segmenting individual leaves in point cloud patches extracted from a field and shows that we achieve accurate predictions even by using point clouds affected by measurement noise and big occlusions (see Fig. 3) and training only on 37 patches. We compare the results of our approach to the state-of-the-art approaches for clustering-based [29] and mask-based [24] point cloud segmentation, to show that our approach has the best performance on segmenting leaves in densely planted breeding plots.

The data of our task is quite different from commonly investigated data from indoor and outdoor environments like the one present in widely adopted datasets like ScanNet [7], S3DIS [1],

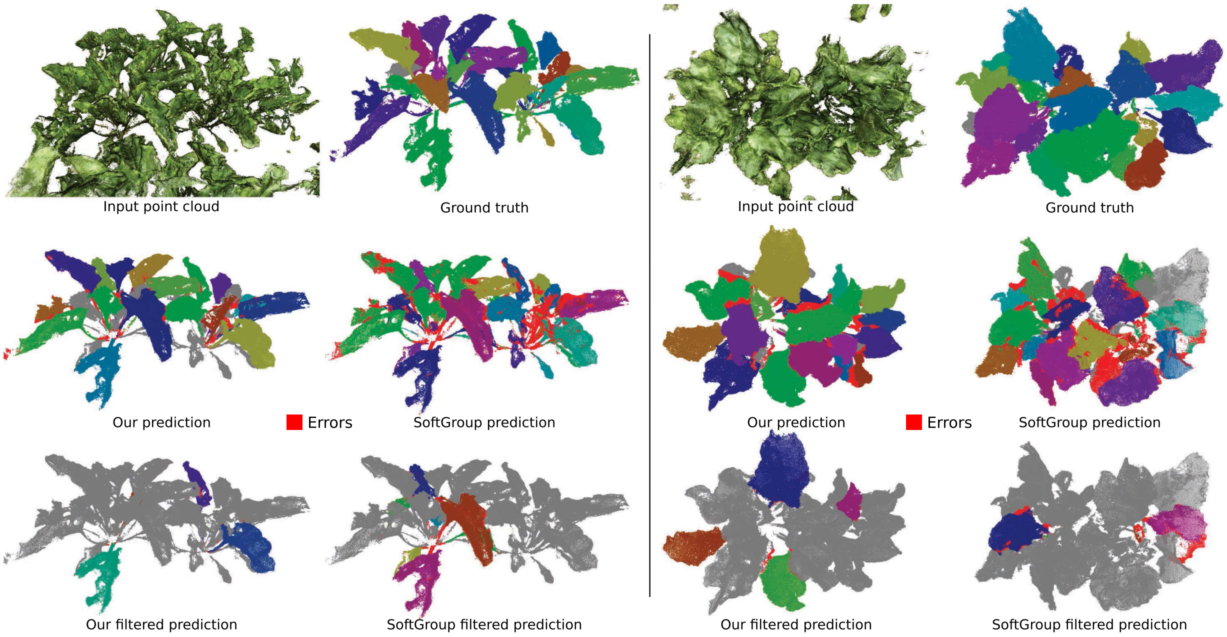


Fig. 4. Qualitative examples of the results of our segmentation approach and SoftGroup. We show the input data and the ground truth in the first row, the full predictions in the middle row and the instances with the highest predicted confidence score in the bottom row. For clarity for the ground truth and predictions we show only the labeled part of the point clouds and the prediction errors are marked in red.

TABLE I
PERFORMANCE OF THE PROPOSED BASELINES AND OUR APPROACH ON SEGMENTING FIELD PATCHES

Approach	PQ [%]	SQ [%]	RQ [%]
Mask3D [24]	18.47	67.71	25.80
SoftGroup [29]	72.07	79.45	90.69
Ours	**75.58**	**80.97**	**93.17**

TABLE II
PERFORMANCE OF THE PROPOSED BASELINES AND OUR APPROACH ON DATA WITH REDUCED NOISE AND FULLY ANNOTATED POINT CLOUDS

Approach	PQ [%]	SQ [%]	RQ [%]
Mask3D [24]	80.28	88.57	90.48
SoftGroup [29]	83.61	89.85	92.98
Ours	**87.10**	**92.06**	**94.54**

or SemanticKITTI [2]. On one hand, the background can be easily segmented by color-based classic approaches, leaving us only with leaves and making the semantic segmentation superfluous. On the other hand, the instance segmentation is challenging as the leaves are heavily overlapping, are affected by high occlusions, and neighbors in breeding plots look very similar as they are of the same variety, making it hard to separate them. Additionally, the point clouds show also substantial amounts of noise due to wind and temperature induced deformations of the plants during the data collection (see Fig. 3).

For the comparison, we use panoptic quality (PQ) [14] defined as follows:

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}|}}_{\mathrm{SQ}} \underbrace{\frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}}_{\mathrm{RQ}}. \tag{13}$$

As can be seen in Table I, we outperform the baseline approaches in all metrics. Mask3D's [24] performance suffers a lot from the fact that labels are not present for all points in the point cloud patches while training. In their approach, instances that are not assigned to a ground truth label are regarded as not being an object, which in the unlabeled parts leads to inconsistencies. SoftGroup [29] instead cannot cope well with the point clouds that contain many erroneous points due to noise in the data and makes segmentation mistakes, especially in the lower parts of the plants. Some exemplary results can be seen in Fig. 4.

D. Leaf Instance Segmentation on Presegmented Plants

The second experiment shows the performance of our approach in segmenting individual leaves in plants that have been manually annotated (for details see Section IV-A). Our aim here is to evaluate our approach in a setting that is closer to the type of data the baselines were developed for. While still training and evaluating on agricultural data, the samples used in this experiment have annotations for all points and most of the noise has been manually removed during the labeling process. Especially the noise due to the deformation of the plants, which shows as multiple occurrences of the same leaf in different positions, shown in Fig. 3(d), is challenging to tackle and it is not present in this experiment.

As expected the results shown in Table II improve on this data. Mask3D improved drastically backing the assumption that the low performance in the previous experiment was due to noise and partial labels. Our approach gains improvements on the results as well, leading to a slightly better performance compared to the baselines also in this simple setting.

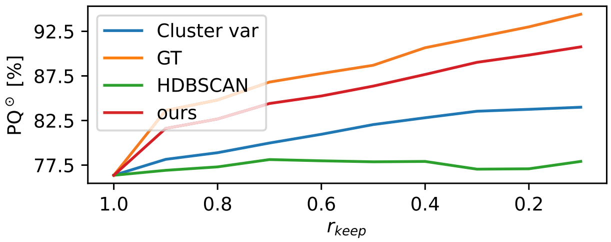


Fig. 5. Evolution of PQ$^\odot$ metric with varying values of r_{keep}. The monotonic increase of our results shows that the confidence is succesfully predicting the quality of the predictions. We also show the upper bound of the $PQ^\odot$ metric computed by filtering the prediction based on the ground truth IoU values.

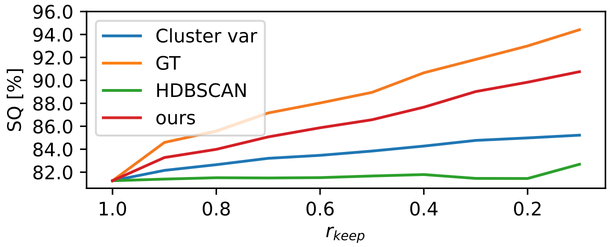


Fig. 6. Evolution segmentation quality (SQ) with varying values of r_{keep}. We outline this metric explicitly as a good value is necessary for the correct estimation of size based phenotypic features.

E. Prediction Accuracy Estimation

The last experiment evaluates our confidence network and shows that our approach is capable of estimating meaningful confidence values for the leaf instance predictions, allowing us to filter out the worst leaf predictions. To evaluate the performance in such a setting we defined the metric PQ$^\odot$ $=$ SQ $\cdot$ PR, as the product of the segmentation quality SQ as defined in (13) and the precision PR.

PQ$^\odot$ therefore accounts for the accuracy of the predictions (SQ) and for precision of the detected instances (PR). It therefore ignores missed predictions (false negatives) as in the targeted application recall is important, as long as it is above the application specific threshold [6]. Both, Mask3D [24] and SoftGroup [29] output a confidence score along with the instance predictions. We used those values to filter the predictions of those approaches.

To filter the predictions of our method we use the confidence estimates computed as explained in Section III-B. As comparison we also evaluated the cluster probabilities of HDBSCAN [19] and the variance of the leaf center predictions $\sigma^2_{lc,\,i}$ as confidence estimates. We define the variance of the leaf center predictions for leaf $\mathcal{S}_i$ as

$$\sigma^2_{lc,\,i} = \frac{\sum_j((\boldsymbol{p}_j + \boldsymbol{v}_j) - \boldsymbol{c}_{pred})}{|\mathcal{S}_i|}, \qquad (14)$$

where

$$\boldsymbol{c}_{pred} = \frac{\sum_j(\boldsymbol{p}_j + \boldsymbol{v}_j)}{|\mathcal{S}_i|}, \qquad (15)$$

with $\boldsymbol{p}_j \in \mathcal{S}_i$ and $\boldsymbol{v}_j \in \mathcal{V}_i$, where $\mathcal{V}_i$ is the set of predicted offset vectors for the points in $\mathcal{S}_i$. As an upper bound for the improvement of the PQ$^\odot$ metric by filtering by the confidence estimates, we filtered the predictions with the IoU$_{gt}$ values. As we report in Table IV, our confidence prediction network achieves the best performance in all but the recall metric (RC) where SoftGroup is slightly better but our result of 22.2% still leads to the evaluation of many more leaves compared to currently used practices [6]. We show the influence of the ratio of kept leaf instances r_{keep} on PQ$^\odot$ and SQ in Figs. 5 and 6.

One concern that may arise about filtering the predicted instances is a resulting change in the leaf size distribution. This would negatively impact phenotyping applications where this distribution is relevant. Therefore, we evaluated how much the distribution of the predictions by our segmentation network changes after filtering with the two most promising approaches, our network and the cluster variance σ^2_{lc}. For this, we use the Kullback-Leibler divergence (KL-div) and the χ^2 test and reported the results in Table III. For clarity, we also show the original and the filtered distributions in Fig. 7. This experiment shows that our approach can accurately predict the quality of predicted leaf instances, while keeping the leaf size distribution almost unchanged.

TABLE III
EFFECTS OF FILTERING ON THE LEAF SIZE DISTRIBUTION

Approach	KL-div	χ^2
Cluster var σ^2_{lc}	0.39	0.68
Ours	**0.28**	**0.57**

We report the kullback-leibler divergence and the chi square test between leaf size distribution before and after filtering.

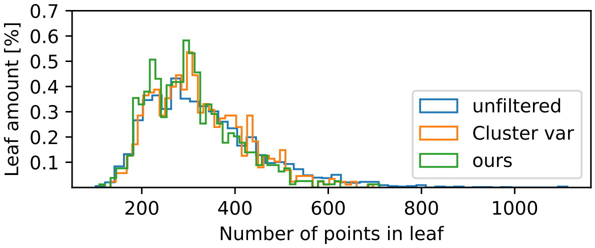


Fig. 7. Both filtering methods change the unfiltered leaf size distribution (blue) only very slightly. For quantitative results see Table III.

F. Ablation on Feature Input for Confidence Network

To evaluate the influence of our proposed feature input, we perform an experiment comparing two setups. The main setup is the one used in our final approach, consisting in inputting the 3D offset of the center prediction of each point from the mean center prediction for the leaf as features for all points that are predicted to be part of the leaf. For the points that are part of the neighborhood of the current leaf, we instead pass zero vectors as features, see (4). In the baseline setup for comparison, we instead pass one-vectors for the points contained in the leaf instance and zero-vectors for the other points, see (5). As metrics for the comparison we use the mean absolute error (MAE) of the IoU prediction and the mean overestimation MO $= \sum_i {}^{oe}/_{|\delta|}$ with

$$oe = \begin{cases} \delta_i - \delta_i^*, & \text{if } \delta_i - \delta_i^* > 0 \\ 0, & \text{otherwise,} \end{cases} \qquad (16)$$

where δ_i and δ_i^* are predictions and ground truth values.

TABLE IV
PERFORMANCE OF THE BASELINES AND OUR APPROACH AT FILTERING OUT BAD PREDICTIONS BASED ON THE CONFIDENCE PREDICTION

Approach	$\text{PQ}^{\odot}_{10}$ [%]	SQ [%]	PR [%]	RC [%]
Mask3D [24]	18.30	59.03	31.00	10.18
SoftGroup [29]	85.34	86.59	98.55	**25.26**
Ours + HDBSCAN [19]	77.92	82.69	94.23	20.83
Ours + CV	83.99	85.22	98.55	18.29
Ours + Confidence	**91.32**	**91.32**	**100.00**	22.23

TABLE V
EFFECT OF USING OUR FEATURES INSTEAD OF INSTANCE MASKS AS INPUT FEATURES TO THE CONFIDENCE NETWORK

Input features	IoU MAE [%]	IoU overestimation (MO) [%]
Instance mask	7.75	4.95
Ours	**6.93**	**4.53**

In the results that we report in Table V, it can be seen that using our features improves the IoU estimation performance and also slightly decreases the overestimation.

V. CONCLUSION

In this letter, we present a novel approach to segment leaves from point clouds acquired using cameras in real field conditions to support plant phenotyping. We show that compared to existing works our approach delivers improved performance in the presence of noise and occlusions, which is a serious challenge in basically all real world applications in crop fields. Additionally, we proposed a novel approach to identify low accuracy predictions in order to discard them, which improves the segmentation performance of a subset of the leaves by a good margin, more specifically 13 percent points on the $\text{PQ}^{\odot}$ metric. In sum, the complete pipeline enables us to obtain a highly accurate segmentation of a subset of the leaves in a plot, enabling the extraction of many plant and leaf phenotypic traits.

REFERENCES

[1] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.

[2] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[3] E. Castro, J. S. Cardoso, and J. C. Pereira, "Elastic deformations for data augmentation in breast cancer mass detection," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inform*, 2018, pp. 230–234.

[4] B. Cheng et al., "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12475–12485.

[5] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3075–3084.

[6] E. Commission, "Protocol for tests on distinctness, uniformity and stability," *Beta Vulgaris l. ssp. Vulgaris var. Altissima Döll.*, 2018.

[7] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[8] F. Fiorani and U. Schurr, "Future scenarios for plant phenotyping," *Annu. Rev. Plant Biol.*, vol. 64, pp. 267–291, 2013.

[9] R. Furbank, J. Jimenez-Berni, B. George-Jaeggli, A. Potgieter, and D. Deery, "Field crop phenomics: Enabling breeding for radiation use efficiency and biomass in cereal crops," *New Phytologist*, vol. 223, no. 4, pp. 1714–1727, 2019.

[10] R. Guo, L. Qu, D. Niu, Z. Li, and J. Yue, "LeafMask: Towards greater accuracy on leaf segmentation," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2021, pp. 1249–1258.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[12] K. Heiwolt, T. Duckett, and G. Cielniak, "Deep semantic segmentation of 3D plant point clouds," in *Proc. Conf. Towards Auton. Robotic Syst.*, 2021, pp. 36–45.

[13] L. Jiang, H. Zhao, S. Shi, S. Liu, C. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4867–4876.

[14] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9404–9413.

[15] F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss, "Towards in-field phenotyping exploiting differentiable rendering with self-consistency loss," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13960–13966.

[16] F. Magistri et al., "Contrastive 3D shape completion and reconstruction for agricultural robots using RGB-D frames," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10120–10127, Oct. 2022.

[17] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss, "Mask-based panoptic LiDAR segmentation for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 1141–1148, Feb. 2023.

[18] E. Marks, F. Magistri, and C. Stachniss, "Precise 3D reconstruction of plants from UAV imagery combining bundle adjustment and template matching," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2259–2265.

[19] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2017, pp. 33–42.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[21] G. Roggiolani, F. Magistri, T. Guadagnino, G. Grisetti, C. Stachniss, and J. Behley, "On domain-specific pre-training for effective semantic perception in agricultural robotics," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023.

[22] G. Roggiolani, M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss, "Hierarchical approach for joint semantic, plant instance, and leaf instance segmentation in the agricultural domain," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023.

[23] G. Roggiolani, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss, "Unsupervised pre-training for leaf instance segmentation in 3D," *Under Rev.*, 2023.

[24] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3D for 3D semantic instance segmentation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023.

[25] D. Schunck et al., "Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis," *PLOS ONE*, vol. 16, no. 8, pp. 1–18, 2021.

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

[27] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[28] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.

[29] T. Vu, K. Kim, T. Luu, T. Nguyen, and C. Yoo, "Softgroup for 3D instance segmentation on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2708–2717.

[30] J. Weyler, F. Magistri, P. Seitz, J. Behley, and C. Stachniss, "In-field phenotyping based on crop leaf and plant instance segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2725–2734.

[31] J. Weyler, J. Quakernack, P. Lottes, J. Behley, and C. Stachniss, "Joint plant and leaf instance segmentation on field-scale UAV imagery," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3787–3794, Apr. 2022.