Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Plant-part segmentation using deep learning and multi-view vision

Weinan Shi [a,b], Rick van de Zedde [b], Huanyu Jiang [a,**], Gert Kootstra [b,*]

[a] Department of Biosystems Engineering and Food Science, Zhejiang University, 310058, China

[b] Wageningen University & Research, Wageningen, 6700 AA, the Netherlands

ARTICLE INFO

Article history:
Received 30 April 2019
Received in revised form
18 July 2019
Accepted 23 August 2019
Published online 17 September 2019

Keywords:
digital plant phenotyping
2D images and 3D point clouds
semantic segmentation
instance segmentation

To accelerate the understanding of the relationship between genotype and phenotype, plant scientists and plant breeders are looking for more advanced phenotyping systems that provide more detailed phenotypic information about plants. Most current systems provide information on the whole-plant level and not on the level of specific plant parts such as leaves, nodes and stems. Computer vision provides possibilities to extract information from plant parts from images. However, the segmentation of plant parts is a challenging problem, due to the inherent variation in appearance and shape of natural objects. In this paper, deep-learning methods are proposed to deal with this variation. Moreover, a multi-view approach is taken that allows the integration of information from the two-dimensional (2D) images into a three-dimensional (3D) point-cloud model of the plant. Specifically, a fully convolutional network (FCN) and a masked R-CNN (region-based convolutional neural network) were used for semantic and instance segmentation on the 2D images. The different viewpoints were then combined to segment the 3D point cloud. The performance of the 2D and multi-view approaches was evaluated on tomato seedling plants. Our results show that the integration of information in 3D outperforms the 2D approach, because errors in 2D are not persistent for the different viewpoints and can therefore be overcome in 3D.

1. Introduction

Plant scientists, geneticists and breeders are interested in having a better understanding of the relationship between a plant's genotype, its environment and its resulting plant phenotype (Chen, Chen, Altmann, & Klukas, 2014). A further understanding could provide, for instance, a better insight in which parts of the DNA are related to specific phenotypic traits, and how they play a role in the plant's tolerance to biotic and abiotic stresses. Eventually, this could result in the ability to predict the phenotype from the genotype, allowing the efficient selection of improved cultivars, which is important in our quest for more efficient and sustainable agriculture (Poland & Rife, 2012). Advances in next-generation DNA sequencing techniques, currently allow for the efficient acquisition of vast amounts of accurate genotypic data (Goodwin, McPherson, & McCombie, 2016). In contrast, the

* Corresponding author.
** Corresponding author.

E-mail addresses: wnshi@zju.edu.cn (W. Shi), rick.vandezedde@wur.nl (R. van de Zedde), gert.kootstra@wur.nl (G. Kootstra).

https://doi.org/10.1016/j.biosystemseng.2019.08.014

Nomenclature

∩	Intersection
∪	Union
$\|M\|$	Number of elements in the set M[-]
2D	Two dimensional
3D	Three dimensional
ADAM	Adaptive moment estimation
CNN	Convolutional neural network
d^{node}	Distance between predicted and ground-truth node centres [pixels]
F1	F1-score [-]
FCN	Fully Convolutional Network
FN	False negative [-]
FP	False positive [-]
LUT	Look-up table
M^{pred}	Mask predicted by the network
M^{gt}	Ground-truth mask
Mask R-CNN	Masked region-based convolutional neural network
n	Centre of the ground-truth node area
$\hat{n}$	Centre of the predicted node area
NMS	Non-maximum suppression
ResNet	Residual network
RoI	Region of interest
SGD	Stochastic gradient descent
TP	True positive [-]

commonly used assessments of a plant's phenotype are mainly based on manual assessment, which is expensive, time-consuming, subjective and error-prone. Lacking a vast amount of complete and accurate phenotypic data, in contrast to the readily available genotypic data, this has been called the phenotyping bottleneck.

1.1. *Image-based phenotyping methods*

To bridge the genotyping-phenotyping gap, there is an increased interest in plant phenomics (Furbank & Tester, 2011), and in particular in using image-based digital phenotyping systems to measure morphological traits of plants. Imaging techniques have the advantage of non-destructive detection, high-throughput processing and multi-trait measurement (Li, Zhang, & Huang, 2014). Two-dimensional (2D) image-based phenotyping approaches are common methods to measure morphological plant traits. The PHENOPSIS system, for instance, retrieves information about leaf growth, leaf number, transpiration rate per unit leaf area, and root growth from the images, which are used to study the plant responds to soil-water deficit in *Arabidopsis thaliana* (Granier et al., 2006). Another imaging system, GROWSCREEN (Walter et al., 2007), assesses the growth of seedlings by measuring the leaf area and growth rate based on images from a top-camera. Building upon this system, Jansen et al. (2009) added chlorophyll-fluorescence imaging to measure photosynthetic activity as well as growth for an improved detection of stress tolerance. Minervini, Abdelsamea, and Tsaftaris (2014) proposed a plant-segmentation method to be able to track the growth of multiple plants over time. The disadvantage of these 2D approaches is that the acquired phenotypical traits are not complete, as the methods cannot deal with overlap and occlusion, and not accurate, as size and area measurements are inaccurate, due to the lack of the 3rd dimension.

In order to deal with the limitations of 2D, we advocate phenotyping through three-dimensional (3D) imaging. With the rapid development in computing power and sensor technology, plant traits extracted from 3D models can provide researchers with more profound and accurate information. Different 3D sensing techniques are used to create a 3D model of a plant for phenotyping. Thapa, Zhu, Walia, Yu, and Ge (2018), for instance, used a point cloud acquired with a LiDAR scanner to measure leaf-surface area, leaf inclination angle and the angular distribution of maize and sorghum. Time-of-flight camera have been used in Chaivivatrakul, Tang, Dailey, and Nakarmi (2014), Haiou, Meng, Xiaodan, and Song (2018) and Vázquez-Arellano, Reiser, Paraforos, Garrido-Izard, and Griepentrog (2018) to rapidly get a 3D image of plants. 3D reconstructions of plants were made using 3D laser scanning in Paulus, Behmann, Mahlein, Plumer, and Kuhlmann (2014), Garrido et al. (2015) and Su, Zhu, Huang, and Guo (2018). However, the above 3D methods are sometimes called 2.5D methods, since a point cloud of only one viewpoint is obtained. Pound, French, Murchie, and Pridmore (2014) proposed the combination of multiple viewpoints in order to obtain a full 3D reconstruction of a plant, based on which accurate plant traits, such as the leaf surface area can be determined. To combine the benefits of a full 3D reconstruction with high-throughput efficiency, Golbach, Kootstra, Damjanovic, Otten, and van de Zedde (2015) implemented an efficient shape-from-silhouette method to accurately measure plant volume, stem height and the surface area of individual leaves.

Although a few of the above-mentioned studies measure traits on a plant-organ or plant-part based, most of the methods acquire information on the whole-plant level. To get more detailed phenotypical information, the plant models need to be segmented into the individual organs, such as leaves, stems, and nodes. To advance the state of the art in 2D leaf segmentation, an annotated dataset was made available as part of the leaf-segmentation challenge to train and test leaf-segmentation methods (Minervini, Fischbach, Scharr, & Tsaftaris, 2016). In Scharr et al. (2016), an overview of four methods that joined the challenge is presented. The method proposed by Vukadinovic and Polder (2015) segmented plant from the background using colour and texture features and a neural network and then segmented the leaves using watershed segmentation on the Euclidean distance map. The method proposed by Scharr et al. (2016) combined a superpixel approach and a distance map to segment the leaves and Yin, Liu, Chen, and Kramer (2018) proposed a multi-leaf segmentation and alignment algorithm.

1.2. *Deep-learning based phenotyping methods*

Many of the hand-crafted segmentation methods mentioned above have recently been outperformed by deep-learning methods. Sakurai, Uchiyama, Shimada, Arita, and Taniguchi

(2018), for instance, proposed a fully convolutional network (FCN) (Long, Shelhamer, & Darrell, 2015) for semantic image segmentation, resulting in high performance in plant segmentation. Morris (2018) applied a convolutional neural network (CNN) to detect leaf contours, which combined with watershed segmentation, results in a good leaf-segmentation performance in cluttered images. Ward, Moghadam, and Hudson (2018) applied a masked region-based convolutional neural network (Mask R-CNN) (He, Gkioxari, Dollár, & Girshick, 2017) to segment different instances of leaves, beating other hand-made approaches on the leaf-segmentation challenge. A regression network based on ResNet50 (He, Zhang, Ren, & Sun, 2016) was implemented by Giuffrida, Doerner, and Tsaftaris (2018) for the accurate counting of leaves in images, robust to deal with different plant species.

Despite the great advances in 2D segmentation of plant parts, a clear drawback of the segmentation of 2D images is that it cannot deal with overlap and occlusions. To deal with this, segmentation methods for 3D point-cloud reconstructions of plants are needed. Again, some hand-crafted approaches have been proposed. Li and Tang (2017), for instance, proposed a method that fits the longest possible vertical line from several views on the 3D point cloud, in order to detect the stem of corn plants. After extracting the stem, the leaves remain as clusters in the point cloud. Similarly, Thapa et al. (2018) removed stem points by filtering on the distance to the vertical centre and consecutively applied K-means clustering to segment the points that belonged to each leaf. A Euclidean clustering algorithm including heuristics about the shape of leaves and internodes was used in Nguyen, Slaughter, Max, Maloof, and Sinha (2015) to segment point clouds of cucumber plants. The segmentation of plant parts, however, is often very challenging due to morphological variations of the plant. The assumptions underlying the hand-crafted methods are therefore often violated, resulting in erroneous segmentations. Moreover, the heuristics used hindered the application of the method to different species or cultivars. Therefore, in this paper, the use of deep-learning methods was studied to learn to segment plant parts in 3D.

To our knowledge, no deep-learning method that results in the segmentation of individual plant parts in 3D point clouds of plants exists in the literature. In general, the segmentation of 3D points clouds using deep learning is a fresh field. Some approaches exist, which can be classified into two types. One type of approach is the point-based work that direct deal with unordered 3D point clouds. This includes methods such as PointNet (Qi, Su, Mo, & Guibas, 2017), PointNet ++ (Qi, Yi, Su, & Guibas, 2017), SGPN (Wang, Yu, Huang, & Neumann, 2018) and 3DmFV (Ben-Shabat, Lindenbaum, & Fischer, 2017), which take the 3D point cloud as input and output class labels for every point. The other type of approach is multi-view based, which creates a number of 2D projections from the 3D point clouds, applying deep-learning based segmentation methods on the generated 2D images, and then combining the different projections into a 3D point-clouds segmentation. SnapNet (Boulch, Guerry, Le Saux, & Audebert, 2017), for instance, was used for semantic segmentation of a 3D scene by generating numerous virtual RGB and geometry-encoded images of the 3D scene, training the images through a network, and back-projecting the label predictions to the 3D model to give to each point a label. Similarly, MVCNN (Su, Maji, Kalogerakis, & Learned-Miller, 2015) was used for 3D shape recognition by applying a standard CNN to recognize the shapes in 2D rendered images from a 3D point cloud. Our approach takes inspiration from these multi-view methods, but multi-view 2D camera images were used instead of generated images.

A plant-part segmentation method is proposed based on deep learning in combination with a multi-view camera system, which segments the 2D images and combines the information from multiple viewpoints into a 3D point-cloud representation of the plant. The added value of multi-view 3D segmentation over 2D segmentation is explored. The multi-view camera setup proposed in (Golbach et al., 2015) was used and applied to both a semantic-segmentation network based on FCN (Long et al., 2015) and an instance-segmentation network based on Mask R-CNN (He et al., 2017). Inspired by multi-view methods, the 2D segmentations are combined from different viewpoints to obtain the 3D point-clouds segmentation. A novel 3D voting strategy is proposed and the performance of the system is evaluated on leaf, stem and node segmentation of tomato seedlings. Our approach can overcome the drawbacks of segmentation on 2D images and obtain a promising performance on segmentation of 3D point clouds. The results of the segmentation can be used for trait extraction of a seedling in the future. The reasons to use a multi-view approach is threefold, (a) the method integrates naturally with our multi-camera setup (Golbach et al., 2015), (b) it can benefit from the GPU optimisation of CNNs for 2D segmentation, enabling a future high-throughput system, and (c) this approach can deal with large and variable-sized point clouds, whereas point-cloud-based 3D segmentation methods, such as PointNet++ (Qi et al., 2017), have a limited input size, therefore requiring down sampling with loss of spatial resolution or splitting up in multiple boxes with loss in speed.

2. Materials and methods

2.1. *Overview of the method*

Figure 1 gives an overview of the proposed method. The method performs two types of segmentation; semantic segmentation and instance segmentation. Semantic segmentation is the task of labelling each point in the image with a class, that is, a point-based classification. For example, in this paper, each point is labelled as either background, stem, leaf or node. Semantic segmentation does not cluster the points into different objects and cannot differentiate between instances. Our method also performs instance segmentation, where all points belonging to an object instance are clustered and labelled individually. For instance, our method finds all points belonging to each individual leaf.

Our goal is to perform semantic and instance segmentation on 3D point clouds of seedlings. To this end, our method is based on a multi-view camera system for 3D reconstruction as proposed by Golbach et al. (2015). As shown in Fig. 1, the multi-view 2D images of a seedling are acquired by the system. Next, semantic segmentation and instance segmentation are

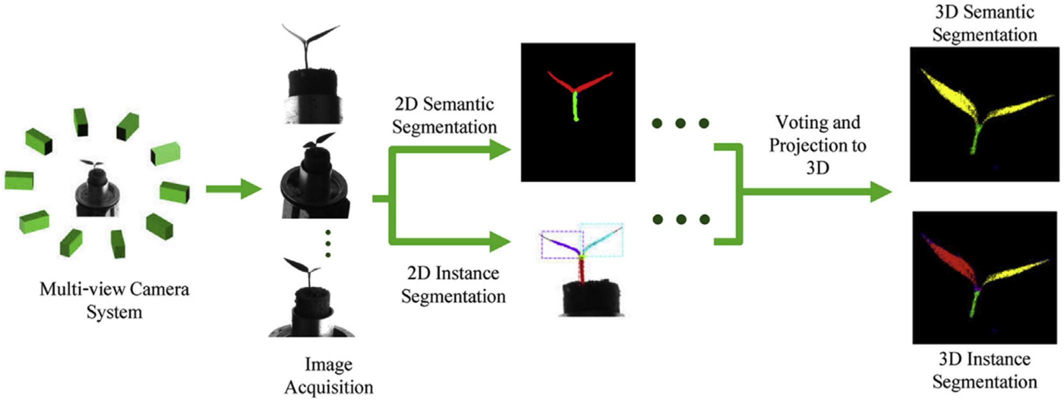

Fig. 1 – Overview of method.

performed on the 2D images. Then, the 2D segmentation results are projected to a 3D point cloud using the known intrinsic and extrinsic parameters of the multiple cameras. Finally, the 3D points are segmented by applying a voting strategy on the results from the multiple 2D segmentations.

In the following subsections, the multi-view camera system is described (section 2.2), as well as the methods for 2D semantic segmentation (section 2.3) and for 2D instance segmentation (section 2.4). Following this, the combination of these results into a multi-view 3D segmentation method (section 2.5) is discussed. The section is completed with a description of the evaluation methods (section 2.6).

2.2. *Multi-view camera system*

The multi-view camera setup as proposed by Golbach et al. (2015) with ten cameras placed in a semi-sphere observing the plant from different viewpoints was used. The cameras were placed at a distance of 900 mm from the seedlings. The system uses Basler acA1300-30 gm cameras, which are affordable cameras that provide grey-scale images with a 1280 × 900-pixel resolution at 30 fps. The use of grey-scale gives a sharper transition between plant and background compared to colour cameras, which use a Bayer pattern. This improves the segmentation of plant and background, resulting in improved 3D point clouds. The plants were segmented from the background using backlighting. The resulting plant silhouettes were input to a shape-from-silhouette method that calculates a 3D voxel representation of the plant combining the silhouettes from all the camera images. With the trade-off between accuracy and speed, the final resolution of the voxel space was set to 0.25 mm/voxel, using 240 × 240 × 300 voxels (x, y, z), resulting in real-world dimensions of 60 × 60 × 75 mm. In this study, we use the individual 2D images as well as the 3D point cloud corresponding to the surface points in the 3D voxel representation of the plant.

2.3. *Semantic segmentation on 2D images*

The goal of semantic segmentation of 2D images of seedlings is to segment the pixels into four classes: background, leaf, stem and node. To meet this need, we use an end-to-end deep-learning approach to semantic segmentation. A fully convolutional network (FCN), as proposed by Long et al. (2015), was used. The outputs of the network are the class labels of each pixel. The used images dataset is discussed followed by the use of FCN to semantically segment the seedling images.

2.3.1. *Dataset for 2D semantic segmentation*

Examples of input images and ground-truth semantic-annotations are shown in Fig. 2. The seedlings in the dataset were in the early stage of development with two leaves. The dataset contains 620 grey-scale images of 62 seedlings with a resolution of 1280 × 900 pixels. The dataset was separated in 420-100-100 for training, validation and testing, respectively. The validation set was used to determine the stopping criterium of training, in order to prevent overfitting. The network was trained until the loss on the validation set stabilized or increased again. The performance evaluation reported in this paper result from the test dataset. Ground-truth pixel-wise annotations were manually obtained using the segmentation tool LabelMe (Russell, Torralba, Murphy, & Freeman, 2008). Pixels were divided into four classes: background, leaf, stem and node. As a large part of the original images contained background, the images were cropped to 600 × 400 pixels before feeding them to the deep neural-network. The different classes were heavily imbalanced, with 99.183% of the pixels being background, while the occupations of leaf, stem and node were 0.600%, 0.207% and 0.013% respectively. During training of the networks, measures were taken to deal with the imbalanced, as will be discussed later.

2.3.2. *Fully convolutional network architecture*

The FCN architecture is shown in Fig. 3. The VGG-16 convolutional network proposed in (Simonyan & Zisserman, 2014), which is commonly used in semantic segmentation tasks was used. As proposed by Long et al. (2015), some modifications at the final layers of the original network were made to accommodate semantic segmentation. The network uses an encoder-decoder structure. In the encoder part of the network, a high-dimensional feature vector is extracted from the image in a series of convolutional layers followed by pooling steps, providing an abstract representation of the image content. The encoder part is illustrated in Fig. 3 by Con1-Pool1- … -Con7. The Con7 layer has a dimensionality of 21 × 13 with 4096 feature channels.

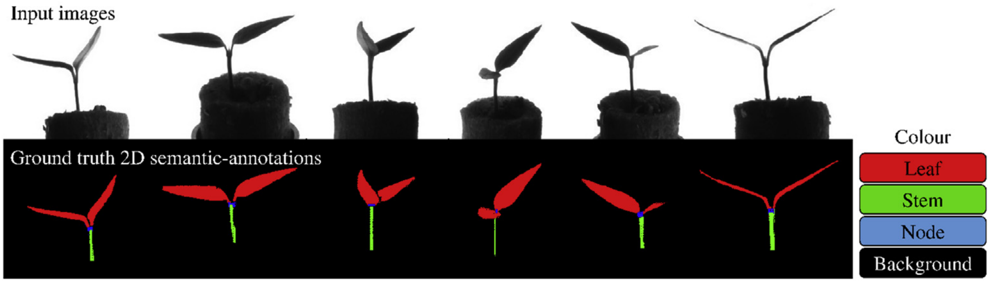

Fig. 2 – Examples of input images (first line) and ground-truth semantic-annotations (second line).

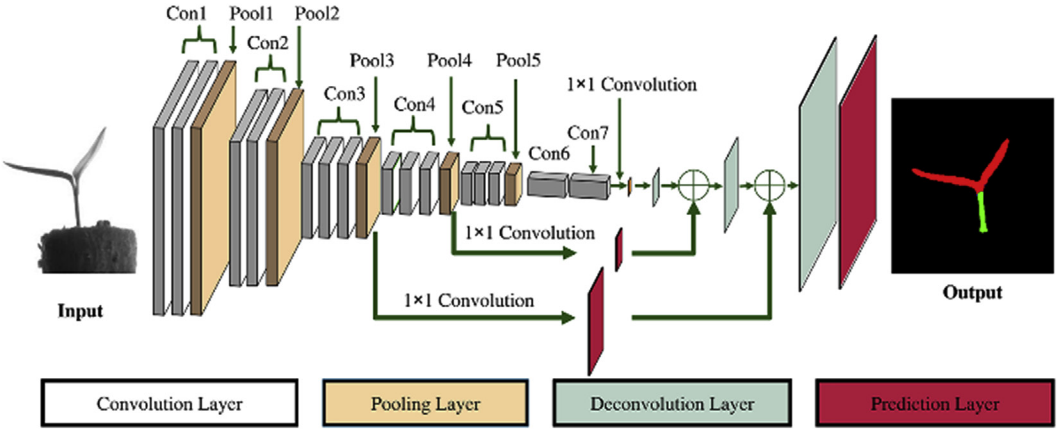

Fig. 3 – Fully convolutional network architecture used for the semantic segmentation. The image refers to the image drawn by Souza (2017) and He et al. (2017).

The decoder part consists of a 1 × 1 convolution layer with 4 (number of classes) channels followed by a series of deconvolution and un-pooling layers to bilinearly up-sample the coarse predictions to pixel-dense outputs of the original resolution. To predict finer details, skip connections were used from the pool4 and the pool3 layer. At the output layer, the network predicts the semantic class of each pixel: background, leaf, stem or node. Transfer learning was employed by copying the pre-trained VGG-16 weights of the first five convolution layers trained on ImageNet (Deng et al., 2009) to our network. The other weights were randomly initialised. The network was trained using stochastic gradient descent (SGD) with weighted cross-entropy loss to handle the imbalanced dataset. The hyper-parameters for training were taken from (Long et al., 2015): a learning rate of 10^{-3}, momentum of 0.9 and weight decay of 5^{-4} for 500 iterations. Dropout was used in the Con6 and Con7 layer.

2.4. *Instance segmentation on 2D images*

Not only is pixel-wise segmentation of the plant images of interest, but also individual plant parts need to be distinguished and segmented; so-called instance segmentation. To this end, MASK-RCNN was employed (He et al., 2017), which is a widely used tool for instance-segmentation tasks. Based on an input image, the network provides a bounding box around the object instance, including a pixel-level segmentation of the object inside the bounding box. The dataset will be discussed, followed by the Mask R-CNN architecture.

2.4.1. *Dataset for 2D instance segmentation*

The same plants and images with the same split in training, validation and test sets as discussed in section 2.3.1 were used. Here, the ground-truth annotations consist of the bounding boxes around the object instances including labels of the instances (e.g. “leaf 1” or “leaf 2”) and a pixel-level segmentation of the instance inside the bounding box (called mask). The stem, node and all individual leaves were also annotated using the LabelMe annotation tool. The train dataset contains 840 instances of leaves, 420 instances of stems and 420 instances of nodes. The test dataset contains 200 instances of leaves, 100 instances of stems and 100 instances of nodes. Figure 4 gives a few examples of the annotated dataset.

2.4.2. *Mask R-CNN architecture*

The Mask R-CNN architecture (He et al., 2017) can be divided into 3 parts: a feature-extraction network, a region-proposal network and two instance detection and segmentation networks, as shown in Fig. 5. The feature-extraction network used had a 50 layer residual network (He et al., 2016), which extracted high-level features from the input image in the final convolutional layer of the 4-th stage (ResNet-50-C4). This is a commonly used network for feature extraction. The region-proposal network proposed a number of initial regions of interest (RoIs) that potentially contain objects of interest (Ren, He, Girshick, & Sun, 2017). The RoIAlign layer is applied to warp the RoIs into fix-sized feature maps. These were then

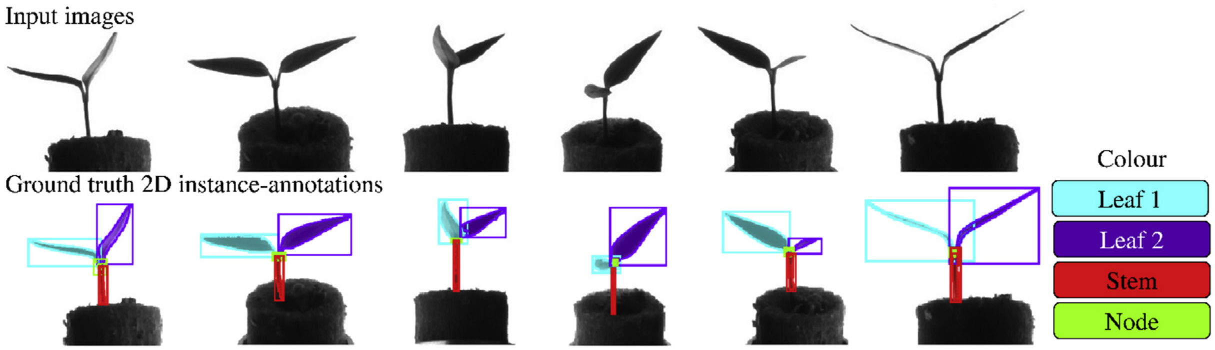

Fig. 4 – Examples of input images (first line) and ground-truth instance-annotations (second line).

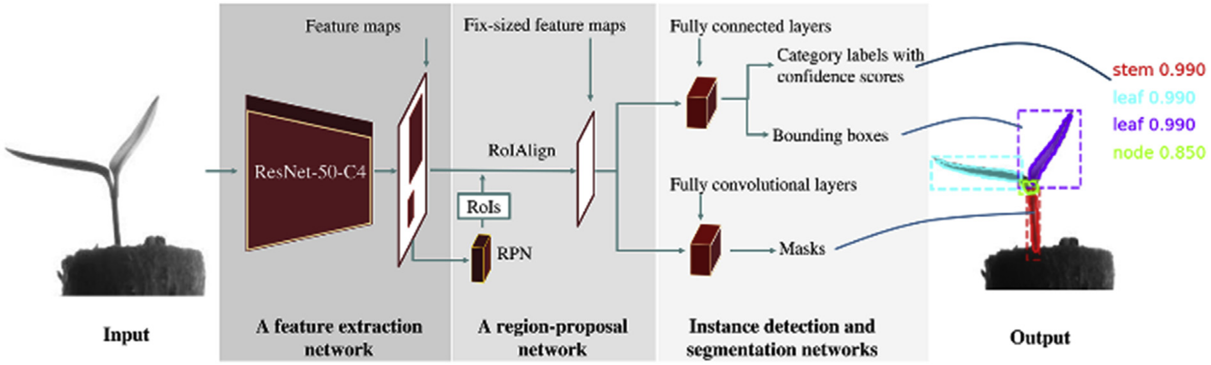
Feature maps
Fix-sized feature maps
Fully connected layers
Category labels with confidence scores
Bounding boxes
Fully convolutional layers
Masks
ResNet-50-C4
RoIAlign
RoIs
RPN
stem 0.990
leaf 0.990
leaf 0.990
node 0.850
Input
A feature extraction network
A region-proposal network
Instance detection and segmentation networks
Output

Fig. 5 – Mask-RCNN architecture used for the instance segmentation. The image refers to the images drawn by (Ren et al., 2017).

input to the instance detection and segmentation network consisting of two branches. One branch was a fully-connected network that takes the RoI as input and detects the presence of an object instance including the bounding box, the class label and the confidence score. The other branch was a fully convolutional network for predicting the pixel mask of the objects within the bounding box. The confidence scores (ranging from 0 to 1) describe how confident the network is about the predicted class.

Hyper-parameter settings are taken from (He et al., 2017), which are set as follows. During training, a RoI was considered correct if it had an intersection-over-union (IoU) with the ground-truth box of at least 0.7. To avoid duplicates for the same object, the non-maximum suppression (NMS) was applied with the threshold set to 0.7. The network was trained using the adaptive moment estimation (ADAM) optimiser for 450 iterations, after which the training loss and validating loss were stable, with a learning rate of 10^{-3} and a weight decay of 10^{-3} for the first 5 iterations, then with a learning rate of 10^{-4} and a weight decay of 5×10^{-4} for the next 30 iterations, and finally at a learning rate of 10^{-5} and a weight decay of 10^{-4} for the final iterations.

2.5. *Segmentation on 3D point clouds*

Segmentation of plant parts in the individual camera images has clear limitations due to occlusions and difficult perspectives. To deal with that, an integration of the image segmentations from different viewpoints into a 3D representation was utilised. The dataset is explained first, followed by a discussion of the multi-view 3D segmentation method.

2.5.1. *Dataset for multi-view 3D segmentation*

To evaluate the performance of the multi-view 3D segmentation method, the 3D point clouds were annotated manually by using Rviz cloud annotation tool (Monica, Aleotti, Zillich, & Vincze, 2017). Each point in the cloud was assigned a class label. In case of instance segmentation, each individual instance got a specific label. Figure 6 contains examples of the annotated point clouds.

2.5.2. *Voting strategy for multi-view 3D segmentation*

In order to segment the 3D point cloud of the plant, the predicted segmentations of all 2D images were combined in 3D using a voting strategy, as illustrated in Fig. 7. The first step is to obtain the intrinsic and extrinsic camera parameters in order to spatially calibrate the multi-view camera system (Golbach et al., 2015). In this procedure the projections of the points in the 3D work space onto each of the ten camera images was determined and stored in look-up tables (LUTs) for computational efficiency. Once the system was calibrated and the 2D-to-3D correspondences were determined, the 3D point cloud of the plant can be obtained using the shape-from-silhouette method as described in section 2.2. Next, for each point in the 3D point cloud, the LUTs were employed to find the corresponding pixel coordinates in the ten camera images. At these coordinates the predicted class label in the 2D images

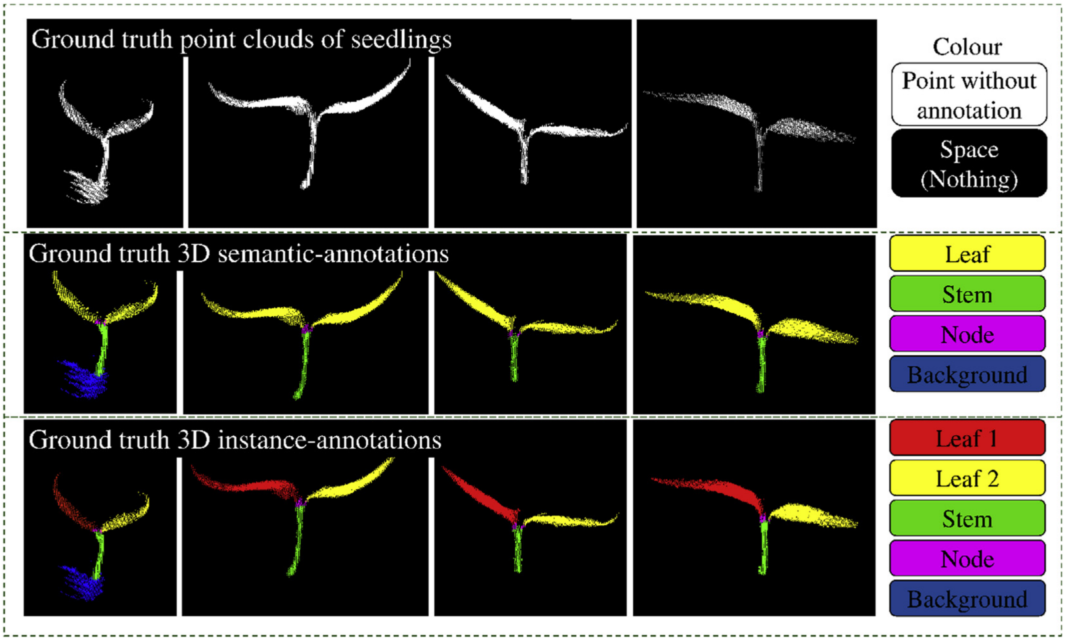
Ground truth point clouds of seedlings
Colour
Point without annotation
Space (Nothing)
Ground truth 3D semantic-annotations
Leaf
Stem
Node
Background
Ground truth 3D instance-annotations
Leaf 1
Leaf 2
Stem
Node
Background

Fig. 6 – Examples of ground-truth point clouds of seedlings, ground truth semantic-annotations and ground truth instance-annotations.

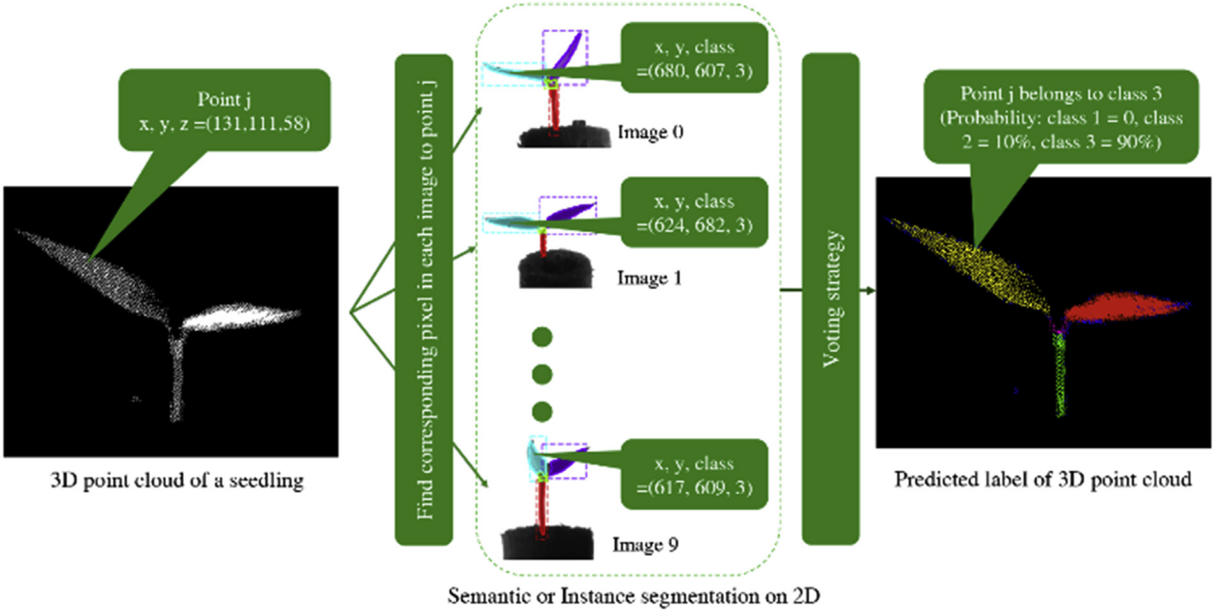
Point j
x, y, z =(131,111,58)
3D point cloud of a seedling
Find corresponding pixel in each image to point j
x, y, class =(680, 607, 3)
Image 0
x, y, class =(624, 682, 3)
Image 1
x, y, class =(617, 609, 3)
Image 9
Semantic or Instance segmentation on 2D
Voting strategy
Point j belongs to class 3 (Probability: class 1 = 0, class 2 = 10%, class 3 = 90%)
Predicted label of 3D point cloud

Fig. 7 – Procedures of projecting 2D results to 3D point cloud. The colors of the bounding boxes and masks on image 0-9 were randomly chosen.

was sampled as predicted by the deep neural networks. Finally, a voting strategy to label the 3D points was used. Each point received ten votes on the predicted labels from the ten images. The probability of each class label was calculated from the proportion of votes. The class label with the highest probability was assigned to the 3D point. The same method was applied for semantic segmentation and for instance segmentation.

2.6. *Performance evaluation*

The performance of the proposed method for plant-part segmentation was evaluated on two levels, on the pixel level and on the object (plant part) level. In both cases, the evaluation was based on precision, recall and F1-score. Precision indicated the proportion of correct detections and recall indicated the proportion of plant parts that were detected by the neural networks. Both measures ranged between 0 (worst) and 1 (perfect):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives. Precision and recall are combined in the F1-score, providing the harmonic average of both:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The difference in the pixel-wise and object-wise evaluation lies in the way that the TP, FP, and FN were calculated. For the pixel-wise evaluation, the annotated ground-truth mask, M^{gt}, of the plant part was compared pixel-by-pixel with the predicted mask, M^{pred}. When both masks agree on the class, the number of true positives is determined by the intersection of the two masks (Eq. (4)). The number of false positives was the number of pixels/points predicted by the network, which were not present in the ground-truth annotation (Eq. (5)). The false negatives were the pixels/points in the ground-truth annotation that were not predicted by the network (Eq. (6)):

$$\text{TP} = \left|M^{\text{pred}} \cap M^{\text{gt}}\right| \tag{4}$$

$$\text{FP} = \left|M^{pred}\right| - \text{TP} \tag{5}$$

$$\text{FN} = \left|M^{gt}\right| - \text{TP} \tag{6}$$

For the object-wise performance evaluation, TP, FP and FN were calculated on the level of the plant parts. To determine if a plant part was detected by the network, the intersection-over-union (IoU) between the annotated ground-truth mask, M^{gt}, and the predicted mask, M^{pred}, both labelled with a particular class, was examined. The IoU ranges from 0 to 1, indicating how well the plant part had been segmented:

$$\text{IoU} = \frac{\left|M^{pred} \cap M^{gt}\right|}{\left|M^{pred} \cup M^{gt}\right|} \tag{7}$$

where $|M|$ indicates the number of pixels/points in the set of points/pixels M. TP is the number of true plant parts that have a corresponding prediction with an IoU ≥ 0.5, FP is the number of predicted plant parts that have no associated true plant part, that is, IoU < 0.5, and FN is the number of true plant parts that were not predicted by the network, that is, IoU < 0.5.

For the node detection, the object-wise evaluation was slightly different. Since the node area was very small, it was difficult to define a precise boundary of a node. This renders the IoU a useless measure for the performance of the node detection. Instead, the Euclidian distance between the centre of the predicted node area, $\hat{n}$, and the center of ground-truth node area, n was used

$$d^{\text{node}} = ||n - \hat{n}|| \tag{8}$$

By setting a distance threshold on d^{node}, the precision, recall and F1-score can be calculated for the node detection. In the evaluation, a threshold of 1 mm and 2 mm was used. If the distance was larger than the threshold, the predicted node was considered as a false positive and the ground-truth was considered as a false negative. If the distance was smaller than the threshold, the predicted node was considered as a true positive. If there was no detection for a node, then the ground truth was considered as a false negative.

To compare the 2D and multi-view 3D segmentation results, Welch's t-test, also called the unequal variances t-test, was applied to test if there is a significant difference in the results of 2D and multi-view 3D segmentation.

3. Results

The focus in the experiments was on the comparison between 2D and 3D plant-part segmentation. In section 3.1, the performance of semantic segmentation was compared and section 3.2 the results for the instance segmentation are shown. In section 3.3, a qualitative analysis on the errors of the method is discussed to provide insights for future improvements.

3.1. *Performance of semantic segmentation*

Figures 8 and 9 show some examples of semantic segmentation on the 2D images and the 3D point clouds respectively using the methods described in section 2.3 and section 2.5. It should be noted that in the 3D case, some points in the background were included in the plant reconstruction due to a reconstruction error made by the shape-from-silhouette method. The deep neural network, however, correctly predicted those points as being background. Table 1 provides the quantitative evaluation of the methods on the complete test set.

In general, the results in Table 1 show that for leaf and stem segmentation, all measures (precisions, recalls, and F1-scores) increase when the 2D segmentation results were combined in the 3D point cloud. Also, the standard deviations on the measures decrease, which indicates that the algorithm's performance was more stable in 3D than in 2D. The p-values resulting from the Welch's t-test show that the improvement was significant. This shows the benefit of the multi-view 3D method. The precision scores of stem and leaf for 2D segmentation were lower than the recall scores. The reason for this is that the predicted area of these plant parts made by the deep neural network was larger than the ground-truth area, which means that the network predicted a portion of background pixels as foreground. In 3D, this is not an issue, due to the shape-from-silhouette method that carves out the plant contours. The precision for stem in 3D was lower than for leaf due to the fact that the stem is a thin structure, consisting of only a small number of points compared to the leaves. The average point number of stem and leaf in each point cloud was 374 and 4029 respectively. Due to the small number of points, false negatives had a strong effect on the precision scores for the stem.

The results for background segmentation could not directly be compared between 2D and 3D. In 2D image segmentation, the measures are all close to 1.00. This is caused by

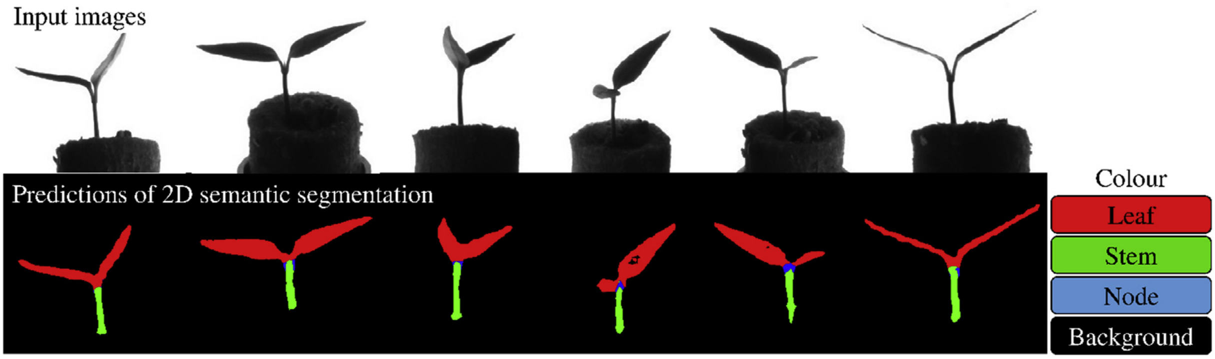

Fig. 8 – Examples of input images (top row) and predictions of semantic pixel labels (bottom row).

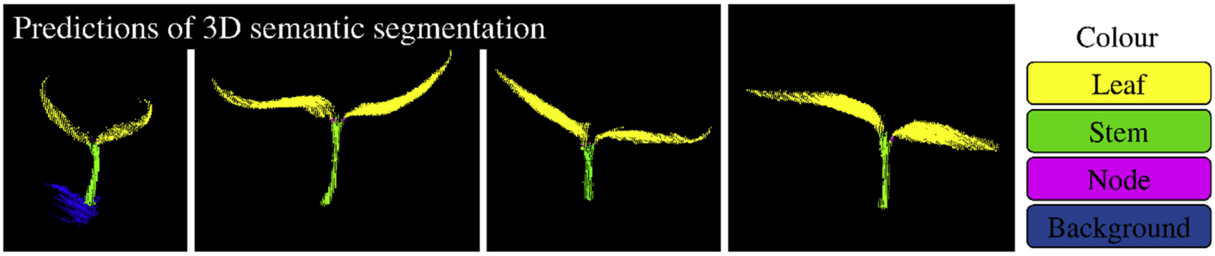

Fig. 9 – Examples of predictions of multi-view 3D semantic segmentation.

Table 1 – Result of semantic segmentation. In the upper part, the performance of the pixel-wise semantic segmentation is shown. The lower part indicates the performance of the object-wise node detection. The results are the averages over respectively the 2D images (N = 100) and the 3D point clouds (N = 60).

	Pixel-wise semantic segmentation								
	Precision			Recall			F1-score		
	2D	3D	P-value	2D	3D	P-value	2D	3D	P-value
Stem	0.54 (0.14)	0.86 (0.09)	0.000[b]	0.90 (0.15)	0.97 (0.05)	0.000[b]	0.68 (0.14)	0.91 (0.05)	0.000[b]
Leaf	0.77 (0.11)	1.00 (0.00)	0.000[b]	0.95 (0.07)	0.99 (0.02)	0.000[b]	0.85 (0.08)	0.99 (0.01)	0.000[b]
Background[a]	1.00 (0.00)	0.54 (0.47)	0.001[b]	1.00 (0.00)	0.89 (0.14)	0.000[b]	1.00 (0.00)	0.85 (0.20)	0.003[b]

	Object-wise semantic detection of node								
	Precision			Recall			F1-score		
	2D	3D	P-value	2D	3D	P-value	2D	3D	P-value
Threshold = 1 mm	0.90	0.82	0.110	0.86	0.65	0.074	0.88	0.73	0.086
Threshold = 2 mm	1.00	1.00	–	0.96	0.93	0.168	0.98	0.96	0.175

The numbers in parentheses are the standard deviations of the results.

[a] Points of background only exist in some point clouds, which results in the low precision in multi-view 3D segmentation. If we calculate the result of background (3D) among the whole dataset, precision and recall will be 0.90 and 0.91, respectively.

[b] There is a significant difference.

the extreme imbalance in the 2D dataset, with the images consisting for 99.183% of background pixels. In the 3D data set, however, there were hardly any background points, as these should be carved out by the shape-from-silhouette method. This explains the low precision score for the multi-view 3D method, as with only a few background points, there was a low number of true positives for the background prediction. Every mistake made (false positive or false negative), therefore drastically reduced the scores. The high associated standard deviations reflect large fluctuations in the scores due to the small number of background points. If the result for background was calculated over the whole dataset, instead of forming an average over the individual point cloud, the precision and recall were 0.90 and 0.91, which means the algorithm seldom detected other instances as background and could detect most background points in 3D point clouds.

The node detection was evaluated with the object-wise performance using the Euclidean distance of the centre of the predicted and ground-truth node area. When the threshold of node distance was set as 2 mm, both nodes in 2D images and 3D point clouds could be well detected with an F1-score at 0.98 and 0.96 respectively. A threshold of 1 mm resulted in an F1-score 0.88 and 0.73 respectively. The p-values indicated that the differences are not significant.

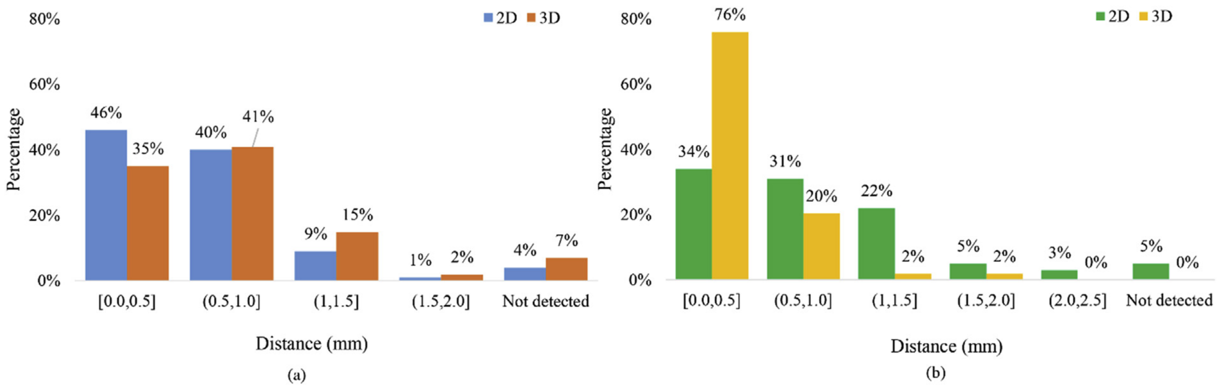

Fig. 10 – Distribution of distance between the centre of predicted node pixels and the centre of ground-truth on (a) semantic segmentation and (b)instance segmentation. 2D represents results of segmentation on 2D images (ground-truth node number in the testing dataset = 100). 3D represents results of segmentation on point clouds (ground-truth node number in testing dataset = 60).

Figure 10a provided more detail on the node detection. It showed the histogram with percentages of real nodes that are detected by the algorithm at particular distances. The plot also shows that for the 2D case, 4% of the nodes were missed. In 3D, 7% of the nodes were missed. The reason for the increasing number of miss detection was that in 2D, only a few pixels were predicted as a node and the location of these points on the plant did not agree in the images from different viewpoints. In the voting procedure, it happened that none of the 3D points got a majority of votes for a node. Hence the node was not detected.

3.2. *Performance of instance segmentation*

Here, the result for 2D and multi-view-3D instance segmentation results by the methods introduced in section 2.4 and section 2.5 are discussed. Some examples are shown in Figs. 11 and 12. The quantitative performance results on the full test set are given in Table 2. Regarding the performance of leaf and stem pixel-wise segmentation, the precision, recall and F1-scores were higher for the multi-view 3D method compared to 2D method, with lower standard deviations. The very low p-values show that the improvement was highly significant. This demonstrates that the multi-view 3D approach was better able to detect the plant parts and provides a more stable detection. The precision scores were particularly high, showing that predictions made by the deep neural network are reliable. The recall in 2D segmentation was relatively low, which means that the algorithm failed to segment some of the plant parts with sufficient overlap between predicted and ground-truth area. The multi-view 3D method was better at segmenting the plant parts as indicated by the higher recall scores. Regarding the object-wise segmentation of leaf and stem, both precision and recall values for the multi-view 3D method were 1.00, indicating that with the IoU-threshold of 0.5, all leaf and stem instance were detected perfectly. In the 2D case, the values are lower. And especially for the leaves, many instances were not detected well. Again, the multi-view 3D method outperformed the method using the individual viewpoints.

The 2D instance-segmentation method only predicted leaf, stem and node instances. The results therefore did not show a performance for the class background. When projecting the 2D results to the 3D point cloud segmentation, some 3D points correspond to pixels in the 2D images for which the neural network did not predict a label. These points were classified as background. In 3D point cloud-segmentation, the F1-score for the background was 0.42 with the precision at 0.17 and the recall at 0.98. These low values, however, were caused by the fact that there were hardly any background points in the 3D point clouds. This results in a low number of true positives and therefore a small number of false positives brings down the score down

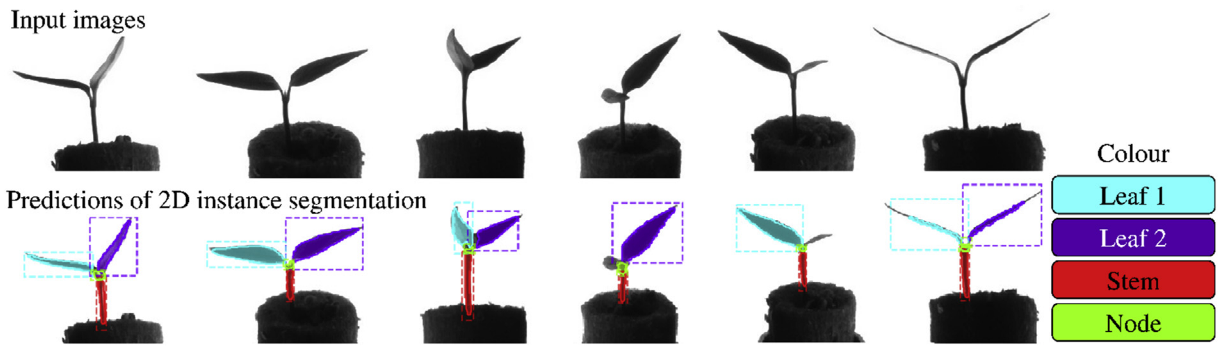

Fig. 11 – Examples of input images (first line) and predictions of instance labels (second line).

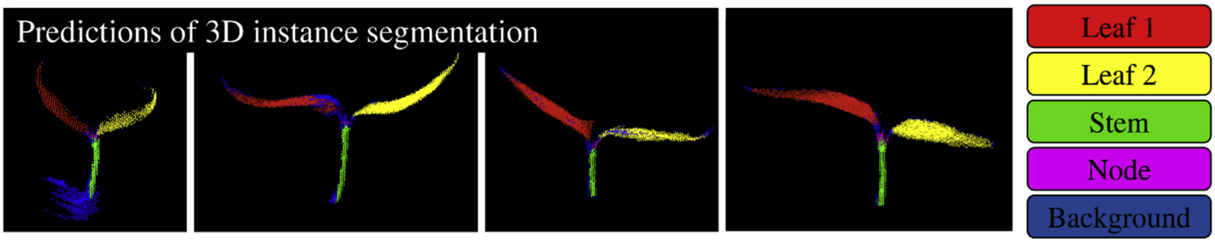

Fig. 12 – Examples of predictions of multi-view 3D instance segmentation.

Table 2 – Results of instance segmentation. In the upper part, the performance of the pixel-wise semantic segmentation is shown. The lower part indicates the performance of the object-wise node detection. The results are the averages over respectively the 2D images (N = 100) and the 3D point clouds (N = 60).

		Pixel-wise instance segmentation								
		Precision			Recall			F1-score		
		2D	3D	P-value	2D	3D	P-value	2D	3D	P-value
Stem		0.77 (0.16)	0.97 (0.04)	0.000[b]	0.65 (0.20)	0.79 (0.09)	0.000[b]	0.70 (0.15)	0.87 (0.06)	0.000[b]
Leaf 1		0.95 (0.10)	1.00 (0.00)	0.000[b]	0.66 (0.19)	0.92 (0.06)	0.000[b]	0.78 (0.14)	0.96 (0.03)	0.000[b]
Leaf 2		0.93 (0.13)	1.00 (0.00)	0.000[b]	0.67 (0.20)	0.89 (0.09)	0.000[b]	0.78 (0.16)	0.94 (0.05)	0.000[b]
Background[a]		–	0.17 (0.26)	–	–	0.98 (0.04)	–	–	0.42 (0.31)	–
		Object-wise instance detection								
		Precision			Recall			F1-score		
		2D	3D	P-value	2D	3D	P-value	2D	3D	P-value
Stem		0.68	1.00	–	0.67	1.00	–	0.68	1.00	–
Leaf		0.83	1.00	–	0.82	1.00	–	0.83	1.00	–
Node	1 mm	0.68	0.96	0.001[b]	0.65	0.96	0.003[b]	0.67	0.96	0.002[b]
	2 mm	0.97	1.00	0.177	0.92	1.00	0.126	0.94	1.00	0.141

The numbers in parentheses are the standard deviations of the results. For the 2D images, no background instance class was considered. In the 3D case, only a small number of background points are in the point cloud, resulting in the low precision score.

[a] Points of background only exist in some point clouds and a large amount of other instance's points is classified as background, which results in the low precision in multi-view 3D segmentation. If we calculate the result of background (3D) among the whole dataset, precision and recall are 0.73 and 0.98, respectively.

[b] There is a significant difference.

severely. This also explains the high standard deviations. If the TP, FN and FP are calculated for background among the whole dataset, instead of the averages over the images, precision and recall were 0.73 and 0.98, respectively. This means that the true background points were detected successfully, but some points on the plant were falsely labelled as background.

Regarding node detection, the multi-view 3D method detected most of the nodes within a distance of 1 mm, with an F1-score of 0.96, whereas the 2D method had a F1-score of 0.67. The results for 3D were significantly better than in the 2D case. When the distance threshold was set to 2 mm, both the 2D and multi-view 3D methods detect the node well, with F1-scores of 0.94 and 1.00 respectively. Although the performance for the multi-view 3D method was better, the difference was not significant. Figure 10b shows the number of detected nodes at different distances, which indicates that the multi-view 3D instance segmentation reliably and accurately detected the nodes, whereas the 2D images missed 5% of the nodes.

3.3. *Qualitative analysis of errors*

In the previous subsections, the quantitative results showing that the proposed multi-view 3D systems for segmentation outperformed the 2D systems were presented. In this subsection, the errors that the systems made are examined in more detail, this is in order to better understand the reasons behind this improvement and to find the limitations of the systems and possibilities for future improvements. In section 3.3.1, the qualitative analysis for 2D and multi-view-3D semantic segmentation is presented, and in section 3.3.2, the qualitative analyses of 2D and multi-view-3D instance segmentation are presented.

3.3.1. *Qualitative analysis of semantic segmentation performance*

The different types of errors made by the 2D semantic-segmentation method are illustrated in Fig. 13. Four different types of error were distinguished:

(a) The segmentation was poor at the borders of plant parts. At the leaf borders, the segment extends into the background, and segmentation errors were made at the junction between leaf and stem. This explains partly the low precision for 2D semantic segmentation. This type of error occurred in every image in the test set (Fig. 13a).

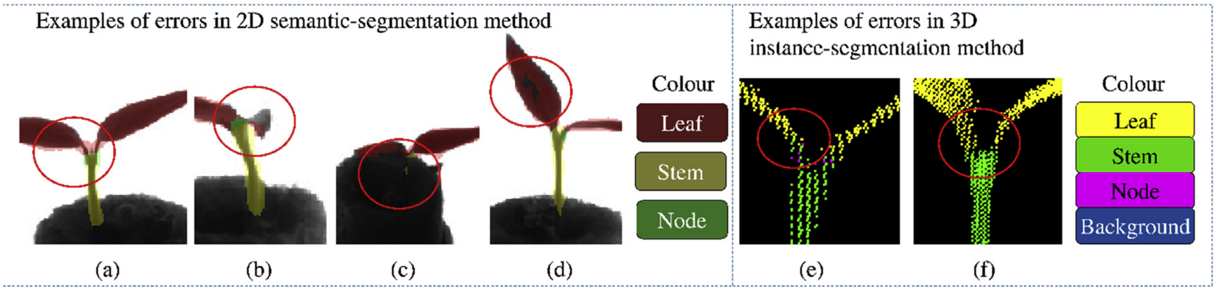
Examples of errors in 2D semantic-segmentation method
Examples of errors in 3D instance-segmentation method
Colour
Leaf
Stem
Node
Colour
Leaf
Stem
Node
Background
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 13 – (a)–(d) provide examples of four types of errors of the 2D semantic-segmentation method: (a) the algorithm has difficulties at the borders of leaves and junction between leaves and stem, (b) the algorithm cannot always deal with abnormal views of the leaves, (c) the algorithm has an incomplete segmentation on areas with a low foreground-background contrast, (d) the algorithm sometimes has an incomplete segmentation within an organ, usually on leaves. (e)–(f) show two types of errors that occasionally occur in the multi-view 3D semantic-segmentation method: (e) the stem segment extends into the leaves and (f) miss detection of the node.

(b) The algorithm cannot deal with abnormal views of the leaves (Fig. 13b), resulting in false negatives, lowering the recall. This error appeared in 3.0% of the images and in total, 4.0% of the images contain abnormal views of the leaves.
(c) The algorithm had an incomplete segmentation on areas with a low foreground-background contrast (Fig. 13c), resulting in false negatives, lowering the recall. This error appeared in 37.0% of the images, while 39.0% of the images contained parts of the seedlings with a low contrast with the background (the plug).
(d) The algorithm had incomplete segmentation within a plant part (Fig. 13d), resulting in false negatives, thus lowering the recall. This happened in 6.0% images.

The multi-view 3D system did not show most of the errors that the 2D system made. Error (a), the over-extension of the segments in the background, was dealt with by the shape-from-silhouette method, which removes background points from the point cloud. Errors (b), (c) and (d) were diminished due to the voting strategy, as these errors occurred at specific places in specific viewpoints but were not consistent for the different viewpoints. The most prominent error remaining in the multi-view 3D method was that some of the leaf points were classified as stem, as shown in Fig. 13e. Because of the low number of stem points, this error resulted in a relatively low precision for stem segmentation. It also lowered the recall for leaf segmentation. Another main error in multi-view 3D semantic segmentation was the miss detection on nodes (Fig. 13f), which occurred in 7% of the testing dataset. This error was discussed in section 3.1.

3.3.2. *Qualitative analysis of instance segmentation performance*

Examples of errors made by the 2D instance-segmentation network can be divided into six specific classes:

(a) The algorithm predicted a mask not related to a plant part (Fig. 14a). This type of error occurred in 0.7% of the instances.
(b) The algorithm predicted multiple masks for one instance (Fig. 14b), influencing the recall of pixel segmentation and resulting in at least one false positive and a false negative in the object-wise evaluation due to low IoU. This type of error occurred in 0.3% of the instances.
(c) The algorithm predicted a mask that only contains a part of the instance (Fig. 14c), resulting in a lower recall in pixel-wise segmentation and a false positive and a false negative in object-wise detection. This type of error occurred in 2.7% of the instances.
(d) The algorithm struggles with viewpoints where the leaves appeared very thin and elongated or very small (<200 pixels area) (Fig. 14d) and where the stems were very small (<200 pixels). This error occurred in 12% of the instances, while 18% of the instances were with such thin or small structures.
(e) The algorithm had poor segmentation on areas with low foreground-background contrast (Fig. 14e), resulting in lower recall for the pixel-wise segmentation and a false positive and a false negative for the object-wise detection 13.7% of the instances were partially in low contrast. This type of error occurred in 8.7% of the instances.
(f) The algorithm missed some of the instances, resulting in lower recall for the pixel-wise segmentation and a false negative for the object-wise detection (Fig. 14f). This error occurred in 1.7% of the instances.

The multi-view 3D systems can avoid most of the errors (a)-(f) that occur in the 2D segmentation due to the voting strategy. Although the multi-view 3D method greatly improved precision and recall over the 2D method, the multi-view 3D method still suffered from a relatively low recall for the pixel-wise segmentation. This was caused by two types of errors: Fig. 14g shows that points near the node area were often classified as background, as well as points near the border of the leaves and the stems, see Fig. 14h. This was caused by the low recalls of leaf and stem in the 2D instance segmentation, classifying part of the plant as background, which occasionally resulted in a majority of votes for

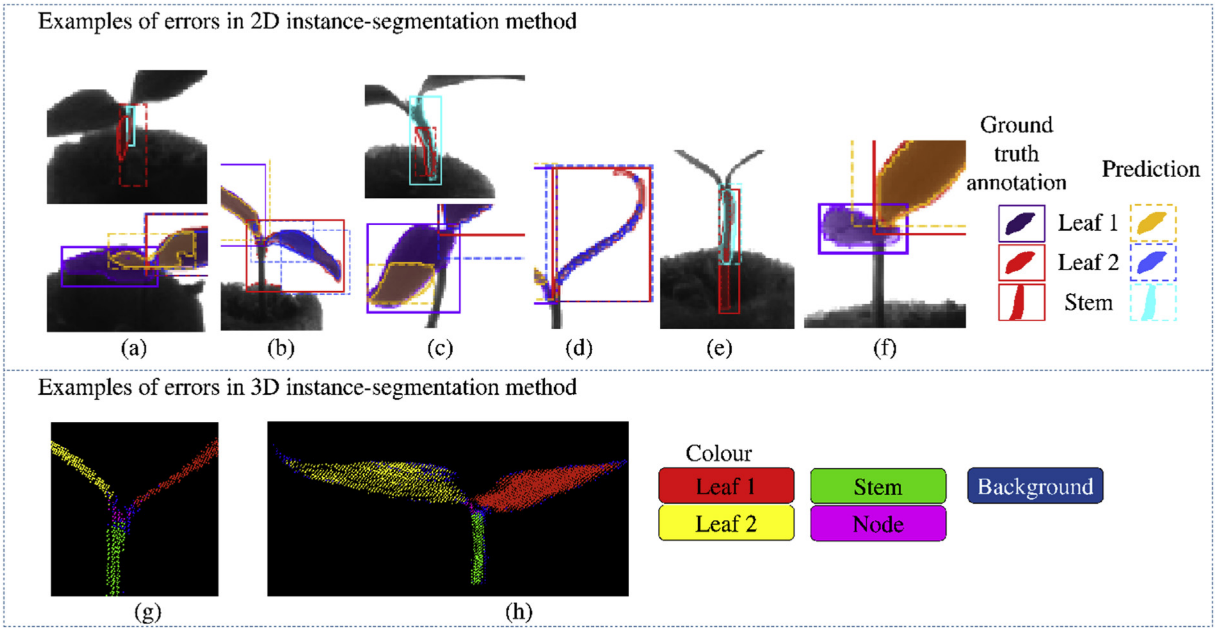

Fig. 14 – (a)–(f) show examples of different types of errors of 2D instance-segmentation method: (a) random prediction, (b) multiple predicted masks for one instance, (c) partial segmentation, (d) bad mask segmentation on thin or small instances, (e) bad segmentation on instances with low foreground-background contrast, (f) miss detection on leaf. (g)-(h) show two types of errors of the multi-view 3D instance-segmentation method: (f) bad segmentation at the node area, and (g) bad segmentations on points at edges.

background on some of the 3D plant points. The errors at the borders of the plant were also an effect of the relatively low resolution of the 3D point cloud with respect to the 2D images. For each 3D point, only one corresponding pixel is considered in each image, whereas in reality it corresponds to an area of pixels.

4. Discussion and conclusion

A deep-learning-based multi-view 3D segmentation method is proposed, which segments 3D point clouds of plants into different plant parts. The method uses a voting strategy to project the segmentation results on 2D images from multiple viewpoints to the 3D point clouds. Two deep-learning networks, FCN for semantic segmentation and Mask-RCNN for instance segmentation were used. The 2D and multi-view 3D method were evaluated on tomato seedlings with the task of segmenting background, leaves, stems and nodes. The methods were evaluated both on the ability to segment the plant on the pixel-level, as well as on object-wise detection. From the results, it could be concluded that the multi-view 3D approach greatly outperforms the 2D approach both for semantic, as well as for instance segmentation. The qualitative analysis identified a number of errors made by the 2D segmentation methods. However, errors that were made in the 2D semantic and instance segmentation did not occur persistently in all viewpoints on the plant. The projection of the segmentation into 3D space and the consecutive voting process result in a majority vote for the classification of a point in which most of the errors were ruled out. On 2D images, segmentation errors were regularly observed, such as bad segmentation on leaves that appear thin, on instances with low foreground-background contrast, on abnormal viewpoints and on small instances, as well as multiple masks and partial segmentation for one instance. However, by projecting the results on the 2D images into a 3D point cloud, the proposed method made up for these errors, as they did not occur for all viewpoints.

The ability to segment different plant parts in 3D enables the estimation of several phenotypic traits (Golbach et al., 2015). The semantic segmentation can provide general traits, such as total leaf area, stem length and plant volume. The instance segmentation can provide additional information on the individual plant parts, such as stem length and diameter, leaf area, dimensions, shape and angle, and internode lengths. In the future, this will provide plant scientists, geneticists and breeders better tools to understand the relationship between genotype, environment and phenotype, and to accelerate breeding.

The used multi-camera setup works under controlled and diffuse illumination, reducing the effects ambient light, shadows and specular reflections. However, the used deep-learning methods have been often used under uncontrolled illumination, and we therefore expect no difficulties for multi-view 3D segmentation in uncontrolled environments.

We proposed a multi-view approach, using CNNs for 2D segmentation and a voting scheme to combine the segmentations from different viewpoints in 3D. This voting scheme assumes that every point in the point cloud is visible from all camera images, which is not always the case due to occlusions. Although, our results show that the majority vote provides generally the correct class, it is recommended to analyse the visibility of the points to allow only votes from cameras that can observe them. Such a more elaborated voting will allow to deal with more complex plants providing more occlusions. The multi-view approach does not provide a true end-to-end solution, as the voting scheme is not optimized by the deep-learning methods. In future work, we will compare the multi-view approach to 3D deep-learning methods that work directly on the point cloud.

In this paper, we used tomato seedlings in an early-growth stage. However, the proposed method should also be able to deal with other plant types and in later growth stages. As backbone, we use deep neural networks for the segmentation of plant parts in 2D images. Others have shown that these methods can deal with more complex plant structures with dense foliage and with different plant species. Kuznichov, Zvirin, Honen, and Kimmel (2019) and Ward et al. (2018), for instance, used a Mask R-CNN as a solution for the leaf-segmentation challenge (Minervini et al., 2016). To deal with the issue of small datasets, they proposed a data augmentation method to synthesise photorealistic images. Morris (2018) proposed a fully convolutional pyramid network to discriminate leaf boundaries to segment leaves in dense foliage. These results can be combined with our multi-view method to combine 2D segmentations from different viewpoints into a 3D segmented plant model. Future work needs to show the accuracy of our method on other plant types and in later growth stages.

The qualitative analysis revealed a number of remaining errors in the multi-view 3D segmentation, which we hope to improve in future work by: (a) improving the data by considering colour images instead of grey-scale, which should improve the contrast between plant and background, as well as including more training data, to provide the network with more examples of abnormal views on the plant; (b) improving the 2D segmentation methods, by adding additional skip connections to include more local detail to improve segmentation at the borders of the plant parts; (c) improving the 3D point cloud method by increasing the resolution, which will improve segmentation at the borders. Also, it is currently assumed that a point on the plant is visible in all camera views, which is not true in reality due to occlusion. By including methods well-known in computer graphics such as z-buffering, visibility can be made more realistic; (d) comparing this method to deep-learning approaches that directly work on the 3D point clouds, such as PointNet++ (Qi et al., 2017) and SGPN (Wang et al., 2018).

Acknowledgements

This work was partly supported by National Natural Science Foundation of China [grant number 31870347] and State Scholarship Fund of China.

REFERENCES

Ben-Shabat, Y., Lindenbaum, M., & Fischer, A. (2017). *3D point cloud classification and segmentation using 3d modified fisher vector representation for convolutional neural networks*. arXiv preprint arXiv:1711.08241.

Boulch, A., Guerry, J., Le Saux, B., & Audebert, N. (2017). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*. https://doi.org/10.1016/j.cag.2017.11.010.

Chaivivatrakul, S., Tang, L., Dailey, M. N., & Nakarmi, A. D. (2014). Automatic morphological trait characterization for corn plants via 3D holographic reconstruction. *Computers and Electronics in Agriculture, 109*, 109–123. https://doi.org/10.1016/j.compag.2014.09.005.

Chen, D., Chen, M., Altmann, T., & Klukas, C. (2014). Bridging genomics and phenomics. In M. Chen, & R. Hofestädt (Eds.), *Approaches in integrative bioinformatics: Towards the virtual cell* (pp. 299–333). Berlin, Heidelberg: Springer Berlin Heidelberg.

Deng, J., Dong, W., Socher, R., Li-Jia, L., Kai, L., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Paper presented at the computer vision and pattern recognition, 2009. CVPR 2009. IEEE Conference on*.

Furbank, R. T., & Tester, M. (2011). Phenomics–technologies to relieve the phenotyping bottleneck. *Trends in Plant Science, 16*(12), 635–644.

Garrido, M., Paraforos, D., Reiser, D., Vázquez Arellano, M., Griepentrog, H., & Valero, C. (2015). 3D maize plant reconstruction based on georeferenced overlapping LiDAR point clouds. *Remote Sensing, 7*(12), 17077–17096. https://doi.org/10.3390/rs71215870.

Giuffrida, M. V., Doerner, P., & Tsaftaris, S. A. (2018). Pheno-deep counter: A unified and versatile deep learning architecture for leaf counting. *The Plant Journal, 96*(4), 880–890. https://doi.org/10.1111/tpj.14064.

Golbach, F., Kootstra, G., Damjanovic, S., Otten, G., & van de Zedde, R. (2015). Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. *Machine Vision and Applications, 27*(5), 663–680. https://doi.org/10.1007/s00138-015-0727-5.

Goodwin, S., McPherson, J. D., & McCombie, W. R. (2016). Coming of age: Ten years of next-generation sequencing technologies. *Nature Reviews Genetics, 17*(6), 333.

Granier, C., Aguirrezabal, L., Chenu, K., Cookson, S. J., Dauzat, M., Hamard, P., et al. (2006). PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit. *New Phytologist, 169*(3), 623–635. https://doi.org/10.1111/j.1469-8137.2005.01609.x.

Haiou, G., Meng, L., Xiaodan, M., & Song, Y. (2018). Three-dimensional reconstruction of soybean canopies using multisource imaging for phenotyping analysis. *Remote Sensing, 10*(8), 1206. https://doi.org/10.3390/rs10081206.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Paper presented at the Proceedings of the IEEE international conference on computer vision*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jansen, M., Gilmer, F., Biskup, B., Nagel, K. A., Rascher, U., Fischbach, A., et al. (2009). Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via GROWSCREEN FLUORO allows detection of stress tolerance in Arabidopsis thaliana and other rosette plants. *Functional Plant Biology, 36*(10–11), 902–914. https://doi.org/10.1071/FP09095.

Kuznichov, D., Zvirin, A., Honen, Y., & Kimmel, R. (2019). *Data augmentation for leaf segmentation and counting tasks in Rosette plants*. arXiv preprint arXiv:1903.08583.

Li, J., & Tang, L. (2017). Developing a low-cost 3D plant morphological traits characterization system. *Computers and Electronics in Agriculture, 143*, 1–13.

Li, L., Zhang, Q., & Huang, D. (2014). A review of imaging techniques for plant phenotyping. *Sensors (Basel), 14*(11), 20078–20111. https://doi.org/10.3390/s141120078.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Minervini, M., Abdelsamea, M. M., & Tsaftaris, S. A. (2014). Image-based plant phenotyping with incremental learning and active contours. *Ecological Informatics, 23*, 35–48. https://doi.org/10.1016/j.ecoinf.2013.07.004.

Minervini, M., Fischbach, A., Scharr, H., & Tsaftaris, S. A. (2016). Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters, 81*, 80–89.

Monica, R., Aleotti, J., Zillich, M., & Vincze, M. (2017). Multi-label point cloud annotation by selection of sparse control points. In *Paper presented at the 2017 International conference on 3D vision (3DV)*.

Morris, D. (2018). A pyramid CNN for dense-leaves segmentation. In *Paper presented at the 2018 15th conference on computer and robot vision (CRV)*.

Nguyen, T. T., Slaughter, D. C., Max, N., Maloof, J. N., & Sinha, N. (2015). Structured light-based 3D reconstruction system for plants. *Sensors (Basel), 15*(8), 18587–18612. https://doi.org/10.3390/s150818587.

Paulus, S., Behmann, J., Mahlein, A. K., Plumer, L., & Kuhlmann, H. (2014). Low-cost 3D systems: Suitable tools for plant phenotyping. *Sensors (Basel), 14*(2), 3001–3018. https://doi.org/10.3390/s140203001.

Poland, J. A., & Rife, T. W. (2012). Genotyping-by-sequencing for plant breeding and genetics. *The Plant Genome, 5*(3), 92–102.

Pound, M. P., French, A. P., Murchie, E. H., & Pridmore, T. P. (2014). Automated recovery of three-dimensional models of plant shoots from multiple color images. *Plant Physiology, 166*(4), 1688–1698.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition*.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Paper presented at the Advances in Neural Information Processing Systems*.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision, 77*(1–3), 157–173.

Sakurai, S., Uchiyama, H., Shimada, A., Arita, D., & Taniguchi, R.-I. (2018). Two-step transfer learning for semantic plant segmentation. In *Paper presented at the ICPRAM*.

Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., et al. (2016). Leaf segmentation in plant phenotyping: A collation study. *Machine Vision and Applications, 27*(4), 585–606. https://doi.org/10.1007/s00138-015-0737-3.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Souza, W.d. (2017). *Semantic segmentation using fully convolutional neural networks*. from https://medium.com/@wilburdes/semantic-segmentation-using-fully-convolutional-neural-networks-86e45336f99b.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Paper presented at the Proceedings of the IEEE international conference on computer vision*.

Su, W., Zhu, D., Huang, J., & Guo, H. (2018). Estimation of the vertical leaf area profile of corn (Zea mays) plants using terrestrial laser scanning (TLS). *Computers and Electronics in Agriculture, 150*, 5–13. https://doi.org/10.1016/j.compag.2018.03.037.

Thapa, S., Zhu, F., Walia, H., Yu, H., & Ge, Y. (2018). A novel LiDAR-based instrument for high-throughput, 3D measurement of morphological traits in maize and sorghum. *Sensors (Basel), 18*(4). https://doi.org/10.3390/s18041187.

Vázquez-Arellano, M., Reiser, D., Paraforos, D., Garrido-Izard, M., & Griepentrog, H. (2018). Leaf area estimation of reconstructed maize plants using a time-of-flight camera based on different scan directions. *Robotics, 7*(4), 63. https://doi.org/10.3390/robotics7040063.

Vukadinovic, D., & Polder, G. (2015). Watershed and supervised classification based fully automated method for separate leaf segmentation. In *Paper presented at the the Netherland Congress on Computer Vision*.

Walter, A., Scharr, H., Gilmer, F., Zierer, R., Nagel, K. A., Ernst, M., et al. (2007). Dynamics of seedling growth acclimation towards altered light conditions can be quantified via GROWSCREEN: A setup and procedure designed for rapid optical phenotyping of different plant species. *New Phytologist, 174*(2), 447–455. https://doi.org/10.1111/j.1469-8137.2007.02002.x.

Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ward, D., Moghadam, P., & Hudson, N. (2018). *Deep leaf segmentation using synthetic data*. arXiv preprint arXiv:1807.10931.

Yin, X., Liu, X., Chen, J., & Kramer, D. M. (2018). Joint multi-leaf segmentation, alignment, and tracking for fluorescence plant videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(6), 1411–1423. https://doi.org/10.1109/TPAMI.2017.2728065.