Joint Plant and Leaf Instance Segmentation on Field-Scale UAV Imagery

Jan Weyler Jan Quakernack Philipp Lottes Jens Behley Cyrill Stachniss

***Abstract*—Monitoring of fields and breeding plots is critical for farmers, plant scientists, and breeders. In this process, a key objective is to assess and monitor the growth stages together with the number of individual plants on the field. Traditionally, this in-field assessment is performed manually and thus is limited in temporal and spatial throughput. In contrast, vision-based systems offer the potential to assess these traits frequently in an automated fashion on a large scale. The primary target of these systems is to detect and segment each plant and its leaves since this information directly correlates to the growth stage and allows for detailed monitoring. In this paper, we address the problem of automated, instance-level plant monitoring in agricultural fields and breeding plots. We propose a vision-based approach to perform a joint instance segmentation of crop plants and leaves in breeding plots. We develop a convolutional neural network to determine the position of specific plant keypoints and group pixels to detect individual leaf and plant instances. Finally, we provide a pixel-wise instance segmentation of each crop and its associated leaves based on orthorectified RGB images captured by UAVs. The experimental evaluation shows that our method outperforms state-of-the-art instance segmentation approaches such as Mask-RCNN on this task.**

***Index Terms*—Robotics and Automation in Agriculture and Forestry, Deep Learning for Visual Perception, Object Detection, Segmentation and Categorization**

I. Introduction

An important aspect of crop breeding and agricultural research is monitoring field trials. This process involves a frequent visual assessment of individual plants at several stages. A key objective is to analyze individual leaves of each plant to monitor its vegetative growth stage [8] as it supports breeders and scientists to select plants that show desirable traits to employ them in further experiments. Typically, each experiment is conducted in spatially separated breeding plots that cover entire agricultural fields. However, this in-field assessment is conventionally done manually [4] and thus limited in spatial and temporal throughput.

In contrast, vision-based plant classification systems offer the potential to monitor field trials in an automated fashion based on images of agricultural robots [10] or unmanned aerial vehicles (UAVs) at a large scale more frequently [22]. Several vision-based learning methods have been proposed in the context of automated vegetation classification on agricultural fields [3], [10], [14]. Typically, such methods perform a semantic segmentation to distinguish pixels belonging to crops and weeds [10] or detect individual plants [3]. However, we see a lack of systems that allow a more in-depth analysis per plant and its organs on agricultural fields.

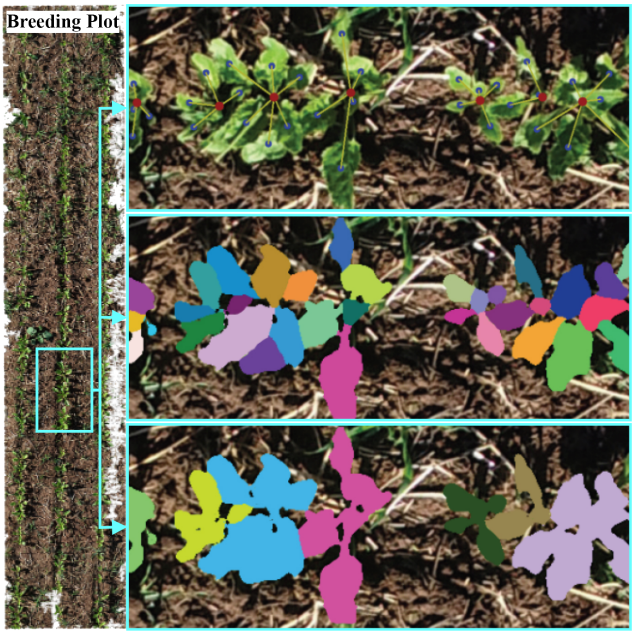

Fig. 1: Left: Orthophoto of a breeding plot. Right: Predictions of our method visualized at a specific location of the plot. Top: Stem keypoints (red), leaf keypoints (blue) and associations (yellow). Center: Pixel-wise mask of each leaf. Bottom: Pixel-wise mask of each plant. We illustrate different instances by different colors.

We propose an approach that predicts a pixel-wise mask of individual crop leaves and assigns each leaf to a particular plant. Consequently, we can access per-plant parameters relevant for field monitoring in an automated fashion, e.g., the total number of leaves per plant and their shape and size, which correlates to vegetative growth stages [8]. This task is challenging since plants may overlap with increasing growth stages, making the association of leaves to a particular plant difficult. Our method offers the potential to perform a high-quality, in-field analysis automatically [22].

The main contribution of this paper is a bottom-up model that provides a pixel-wise instance segmentation of each crop and its associated leaves that can be applied to large-scale

Manuscript received: September 8, 2021; Revised: December 9, 2021; Accepted: January 16, 2022. This letter was recommended for publication by Associate Editor H. Son and Editor Y. Choi upon evaluation of the reviewers' comments. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob and 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) under the innovation support programme under funding no 28DK108B20 (RegisTer).

The authors are with the University of Bonn, 53115 Germany (email: jan.weyler@igg.uni-bonn.de).

Digital Object Identifier (DOI): see top of this page.

orthorectified images of entire fields obtained by UAVs. It provides a georeferenced field-map that allows performing the field analyses as an integrated whole instead of sampling a subset of field locations. In addition, such georeferenced maps enable tracking static objects over time. We propose a single-stage convolutional neural network (CNN) that predicts geometric embeddings, i.e., it determines pixel-wise offsets encoding the association of individual pixels to a leaf or plant stem keypoint. In this context, we define a keypoint as a distinct location of a leaf or plant representative for its associated instance. We employ these embeddings in an automated post-processing step to group individual leaf and plant instances, see Fig. 1.

We make the following three claims. First, our proposed approach detects plant-specific leaf and stem keypoints via predicted offsets. Second, we perform an instance segmentation of crop leaves which we effectively associate to its corresponding plant to conduct a joint instance segmentation of whole sugar beets. Third, on these tasks, we achieve higher performance w.r.t. Mask-RCNN [5] often applied in the agricultural domain [3], [23]. Finally, we published our annotated dataset for comparison: https://www.ipb.uni-bonn.de/data/plis/.

II. Related Work

Semantic scene analysis often relies on CNNs [16], which replaced other learning techniques in this context [12], [24]. Also, there has been an increasing interest in exploiting the potential of UAVs and autonomous ground vehicles [19] for agriculture [17]. These systems target detecting vegetation and localizing individual plant or leaf instances to perform yield prediction, monitoring, and counting.

In an earlier work, Lottes et al. [10] perform semantic segmentation of crops and weeds based on RGB and near infra-red images. Simultaneously, they estimate the stem position of each plant to improve the semantic segmentation and allow for mechanical treatments of weeds. They employ a CNN with two specialized decoders that performs both tasks jointly. In contrast to our approach, this method performs a pixel-wise classification but does not detect instances as required for monitoring breeding plots.

There are competitive approaches to perform instance segmentation, which can be divided into two sets. First, top-down approaches [5], [20] that initially detect bounding boxes for each instance and subsequently generate binary masks for each detected instance, e.g., Mask R-CNN [5]. Second, box-free bottom-up methods [2], [18] that localize keypoints as instance-representatives and jointly map each pixel close to its associated keypoint via predicted offsets. The offsets enforce pixels that belong to the same instance to be close to each other. Thus, they can easily be clustered to perform instance segmentation. Both methods are commonly applied in the agricultural domain [3], [23].

Champ et al. [3] employ Mask R-CNN [5] to perform instance segmentation for crops and weeds on real fields based on RGB images. However, this method does not detect each plant's leaves, which is key information for plant scientists and breeders to monitor breeding plots. Kulikov et al. [7] propose a bottom-up method to detect and segment individual leaf instances based on RGB images that contain single plants. They employ a two-stage method that first specifies target embeddings, which a CNN subsequently learns. At inference, they perform a simple clustering approach to recover each leaf instance. However, this method assumes that each image contains only a single plant which is not realistic for uncontrolled imagery of breeding plots. Magistri et al. [13] propose an automated 4D registration technique based on sequential point clouds of individual plants to track phenotypic traits in laboratory environments.

The aforementioned methods can either perform an instance segmentation of plants or leaves but not both simultaneously. Depending on the particular application, these approaches may be sufficient, e.g., to count the total number of crops on a field. However, for a detailed monitoring of breeding plots a simultaneous instance segmentation of plants and leaves is important to determine relevant parameters for each crop, e.g., the total number of leaves per plant.

Accordingly, Weyler et al. [25] jointly detect the bounding boxes of single plants and per-plant leaf keypoints in images of real fields to compute the total number of leaves per plant. However, this method does not segment individual leaves or plants but provides exclusively coarse keypoints that are unsuitable for determining leaf size and shape. In contrast, we propose a novel approach that predicts the pixel-wise mask of each leaf and associates it with its related plant to provide a more detailed assessment for monitoring.

III. Our Approach

The main objective of our approach is to perform a joint instance segmentation of sugar beet plants and their individual leaves based on RGB images captured by UAVs. Our method works on orthorectified images that we compute through a photogrammetric bundle adjustment [1] using overlapping images covering entire fields. This enables breeders and plant scientists to assess relevant parameters for each plant on the entire field through a single model instead of sampling them at a subset of locations.

We propose a CNN that relies on a simple yet effective topological model of plants. Specifically, we define a plant by its well-defined stem keypoint and a variable number of leaf keypoints, see top of Fig. 1. In this context, we assign each leaf keypoint to a single stem keypoint to model a plant as the union of its leaves. Furthermore, we implicitly associate each keypoint with a unique pixel-wise mask. Particularly, we associate a leaf keypoint with the mask of its related leaf instance and a stem keypoint with the mask of its related plant, see Fig. 1. Thus, we geometrically associate plant pixels to their associated leaf and stem keypoints.

Our network predicts two sets of offset vectors to model the geometric information of each pixel concerning their corresponding leaf and plant instances. One set of offset vectors points for each pixel towards its associated leaf keypoint. We exploit these offsets in an automated post-processing step to first detect the position of leaf keypoints as leaf instance-representatives and then assign each pixel to a specific leaf

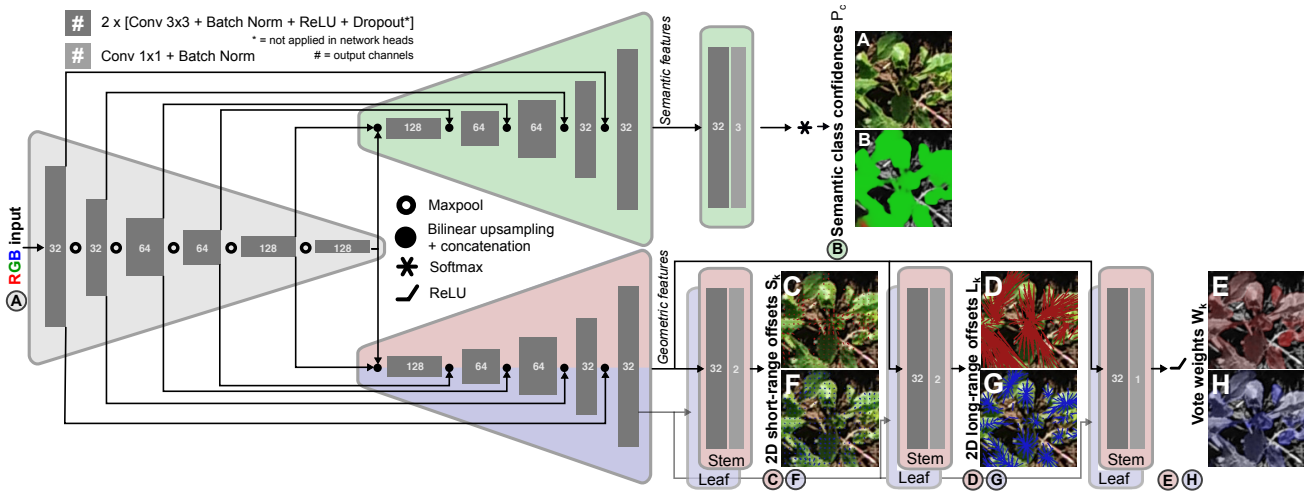

Fig. 2: Our network architecture for semantic segmentation and offset regression with task-specific decoders. We feed the RGB input image $\in \mathbb{R}^{3\times h\times w}$ to a shared encoder and forward its output to separate decoders with h and w being the heigth and width of the image. The upper decoder predicts normalized confidences $P_c \in \mathbb{R}^{3\times h\times w}$ per semantic class $c \in \{\mathsf{beet}, \mathsf{weed}, \mathsf{background}\}$. The lower decoder predicts short-range offsets $\boldsymbol{S}_k \in \mathbb{R}^{2\times h\times w}$, long-range offsets $\boldsymbol{L}_k \in \mathbb{R}^{2\times h\times w}$, and vote weights $W_k \in \mathbb{R}^{h\times w}$ per keypoint type $k \in \{\mathsf{stem}, \mathsf{leaf}\}$. We utilize these predictions to compute crop plant and leaf instances. In the lower left, we show long-range offsets associated with stem (D) and leaf keypoints (G). In addition, we visualize short-range offsets of stem (C) and leaf keypoints (F). Finally, we illustrate the vote weights W_k for stem (E) and leaf (H) keypoints.

to obtain its instance segmentation. The other set of offsets carries the same kind of information, but the offsets do not refer to the leaves but to the stem keypoint of the plants, i.e., the plant instance. To generate the plant segmentation mask, we use the offsets to associate each leaf to a specific stem keypoint in an automated post-processing step. We then generate the instance mask of each plant by joining the corresponding leaf segmentation masks.

A. Ground Truth Annotations

To train our model, we require ground truth data with the following annotations. First, the semantic class label $c(\boldsymbol{x}) \in \{\mathsf{beet}, \mathsf{weed}, \mathsf{background}\}$ per pixel $\boldsymbol{x} \in \mathbb{R}^2$. We treat all pixels with *class* label beet as foreground pixels $\boldsymbol{x}_\mathsf{f}$ and perform the instance segmentation only for those. Second, a sugar beet *instance* label $b(\boldsymbol{x}_\mathsf{f}) \in \mathbb{N}$ and a leaf *instance* label $l(\boldsymbol{x}_\mathsf{f}) \in \mathbb{N}$ per foreground pixel $\boldsymbol{x}_\mathsf{f}$. Third, a unique assignment of each leaf instance l to its associated sugar beet instance b denoted as $b(l) \in \mathbb{N}$. Fourth, a stem keypoint position $\boldsymbol{y}_\mathsf{stem}(b) \in \mathbb{R}^2$ per sugar beet instance b. Finally, a leaf keypoint position $\boldsymbol{y}_\mathsf{leaf}(l) \in \mathbb{R}^2$ per leaf instance l. In contrast to stem keypoints, we set the position of leaf keypoints to the centroid of their mask. Thus, we consider only visible leaves. A domain expert annotated the images at a pixel-level.

B. General Network Architecture

We propose an encoder-decoder network with lateral skip connections [21] that follows a two-branch architecture with two task-specific decoders. Our network shares the encoder and predicts dense feature maps in its task-specific decoders, see Fig. 2. The upper decoder performs a semantic segmentation to determine pixels of interest that belong to sugar beets (Sec. III-C). The lower decoder predicts long-range offsets $\boldsymbol{L}_k(\boldsymbol{x}) \in \mathbb{R}^2$ and short-range offsets $\boldsymbol{S}_k(\boldsymbol{x}) \in \mathbb{R}^2$ per pixel $\boldsymbol{x}$ and keypoint type $k \in \{\mathsf{stem}, \mathsf{leaf}\}$. These offsets translate each pixel of a sugar beet towards its associated leaf and stem keypoint (Sec. III-D). We utilize the predicted offsets for different tasks in an automated post-processing step. First, we estimate the spatial position of leaf and stem keypoints, where each translated pixel casts a vote for its position to be a keypoint. We accumulate these votes in keypoint-specific heatmaps to detect keypoints by a high number of votes. In this context, we predict a voting weight $W_k(\boldsymbol{x}) \in \mathbb{R}$ per keypoint type k and pixel $\boldsymbol{x}$ that considers an object's size to ensure scale-invariance (Sec. III-E). Next, we group sugar beet pixels into individual leaf instances by assigning them to their nearest leaf keypoint via the offsets (Sec. III-F). Third, the offsets serve to associate each leaf to a specific stem keypoint to generate crop plant instances as the union of its leaves (Sec. III-G).

C. Semantic Segmentation

We propose a semantic segmentation branch that computes for each pixel $\boldsymbol{x}$ a probability distribution modeling the assignment to the category beet, weed, or background by their corresponding confidence score $P_c(\boldsymbol{x})$. This allows us to filter pixels with the most probable class weed or background and perform instance segmentation only for sugar beet pixels. However, the segmentation of weeds is still essential to measure weed density.

Consequently, the upper decoder in Fig. 2 predicts a dense feature map for each of the three classes. We apply a softmax activation to obtain the normalized confidence scores for each category. During training, we employ a weighted cross entropy loss [26] based on the predicted scores and ground truth annotations. At inference, we consider a pixel with $P_\mathsf{beet}(\boldsymbol{x}) > 0.8$ as foreground and assign a pixel with $P_\mathsf{weed}(\boldsymbol{x}) > 0.5$ to the class weed. We determined these hyperparameters based on the validation set.

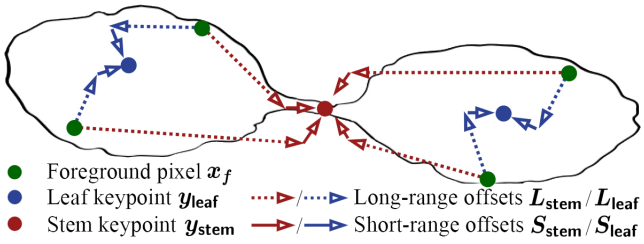

Fig. 3: Visualization of long-range and short-range offsets pointing for each foreground pixel towards its related leaf and stem keypoint.

D. Long-Range and Short-Range Offset

We propose long-range and short-range offsets to translate each foreground pixel precisely towards its associated keypoints, as illustrated in Fig. 3. The long-range offsets define 2D vectors $\boldsymbol{L}_k(\boldsymbol{x}_\text{f}) = \boldsymbol{y}_k(\boldsymbol{x}_\text{f}) - \boldsymbol{x}_\text{f}$, which point for each foreground pixel $\boldsymbol{x}_\text{f}$ towards its associated keypoint $\boldsymbol{y}_k$, where $k \in \{\text{stem}, \text{leaf}\}$. However, the prediction of long-range offsets is known to be inaccurate since they cover large distances [18]. Thus, a translated pixel $\boldsymbol{x}_\text{f}^* = \boldsymbol{x}_\text{f} + \boldsymbol{L}_k(\boldsymbol{x}_\text{f})$ needs further refinement to approach its associated keypoint precisely. Therefore, we define 2D short-range offsets $\boldsymbol{S}_k(\boldsymbol{x}_\text{f}) = \boldsymbol{y}_k(\boldsymbol{x}_\text{f}) - \boldsymbol{x}_\text{f}^*$ to improve the offsets close by the desired keypoint. Finally, we merge both to obtain refined offsets $\boldsymbol{R}_k(\boldsymbol{x}_\text{f}) = \boldsymbol{L}_k(\boldsymbol{x}_\text{f}) + \boldsymbol{S}_k(\boldsymbol{x}_\text{f})$. Since we consider two types of keypoints, we predict two such 2D vector fields for both offsets, shown by four feature volumes in the lower decoder of Fig. 2. The former translates pixels $\boldsymbol{x}_\text{f}$ towards leaf keypoints and the latter towards stem keypoints. During training, we use $L1$ losses for each keypoint type k:

$$\mathcal{L}_k^{\boldsymbol{L}\text{-offsets}} = \frac{1}{|\mathcal{X}_f|} \sum_{\boldsymbol{x}_\text{f} \in \mathcal{X}_\text{f}} \left\| \boldsymbol{x}_\text{f} + \boldsymbol{L}_k(\boldsymbol{x}_\text{f}) - \boldsymbol{y}_k(\boldsymbol{x}_\text{f}) \right\|_1, \tag{1}$$

$$\mathcal{L}_k^{\boldsymbol{R}\text{-offsets}} = \frac{1}{|\mathcal{X}_f|} \sum_{\boldsymbol{x}_\text{f} \in \mathcal{X}_\text{f}} \left\| \boldsymbol{x}_\text{f} + \boldsymbol{R}_k(\boldsymbol{x}_\text{f}) - \boldsymbol{y}_k(\boldsymbol{x}_\text{f}) \right\|_1, \tag{2}$$

where $\mathcal{X}_\text{f}$ is the set of all foreground pixels. We minimize the objective in Eq. (1) to optimize the predicted long-range offsets explicitly. In Eq. (2) we optimize both offsets jointly to enforce our network explicitly to predict large offsets for $\boldsymbol{L}_k(\boldsymbol{x}_\text{f})$ and small offsets for $\boldsymbol{S}_k(\boldsymbol{x}_\text{f})$. Finally, we average the losses for both keypoint types k during optimization.

E. Keypoint Detection

At inference time, we employ the predicted offsets to obtain the instance relationships via an automated post-processing step. First, we exploit the refined offsets to estimate the spatial position of keypoints as instance-representatives via a voting scheme [18]. Since the offsets point for each foreground pixel towards their related keypoint, we aggregate a heatmap where each pixel $\boldsymbol{x}_\text{f}$ translated by $\boldsymbol{R}_k(\boldsymbol{x}_\text{f})$ casts a vote to its position. Accordingly, a high number of votes indicates the position of a keypoint, see Fig. 4. We consider two keypoint-specific heatmaps to extract the position of leaf and stem keypoints separately. The former employs $\boldsymbol{R}_\text{leaf}(\boldsymbol{x}_\text{f})$ and the latter $\boldsymbol{R}_\text{stem}(\boldsymbol{x}_\text{f})$. We use bilinear interpolation to distribute votes into discrete cells.

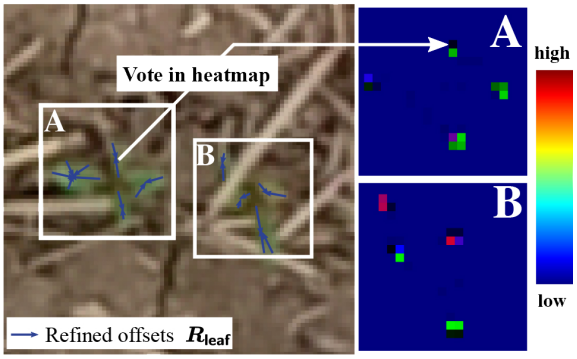

Fig. 4: Left: Image patch of sugar beets at early growth stages and refined offsets pointing towards leaf keypoints. Note that we show only a few offsets for reasons of clarity. Right: Cropped patches from the heatmap for leaf keypoints obtained by our voting scheme.

However, we must assign a voting weight to each pixel $\boldsymbol{x}_\text{f}$ to prevent large instances from dominating small ones in the heatmaps. Thus, we predict a weight $W_k(\boldsymbol{x})$ per keypoint type k and pixel $\boldsymbol{x}$ that contains large weights for pixels whose related instance is small and vice versa, see Fig. 2.

We use a L1 loss for each predicted W_k during training:

$$\mathcal{L}_k^W = \frac{1}{|\mathcal{X}_f|} \sum_{\boldsymbol{x}_\text{f} \in \mathcal{X}_f} \left\| W_k(\boldsymbol{x}_\text{f}) - \frac{\kappa}{A_k(\boldsymbol{x}_\text{f})} \right\|_1. \tag{3}$$

Let $A_k(\boldsymbol{x}_\text{f})$ be the area of the leaf or plant instance associated with the pixel $\boldsymbol{x}_\text{f}$ and κ be a hyperparameter set to $100\,\text{px}^2$. By minimizing Eq. (3), we enforce high weights for pixels associated with small instances and vice versa. Finally, we average the losses over both keypoint types for optimization.

At inference, we recover the position of keypoints by considering local maxima in the heatmaps as keypoint candidates $\boldsymbol{y}_k^c$, see Fig. 4. However, we suppress lower-scoring candidates within a certain radius $\delta_k(\boldsymbol{y}_k^c)$. Intuitively this radius should be small for small instances to avoid the suppression of actual keypoints lying close together and larger for large instances to reduce false positive detections. Thus, we first approximate the extent of an instance for the given candidate by accounting for the set of all pixels $\mathcal{X}_{\boldsymbol{y}_k^c}$ voting for the candidate:

$$\lambda_k(\boldsymbol{y}_k^c) = \sqrt{\frac{\kappa}{|\mathcal{X}_{\boldsymbol{y}_k^c}|} \sum_{\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{y}_k^c}} \frac{1}{W_k(\boldsymbol{x})}}. \tag{4}$$

This value increases as the total number of pixels voting for this candidate increases, i.e., large instances and vice versa. Next, we suppress all lower scoring candidates within the radius $\delta_k(\boldsymbol{y}_k^c) = \min(\frac{1}{2}\lambda_k(\boldsymbol{y}_k^c), \lambda_\text{max})$, where λ_max is the maximum rejection radius, here set to $15\,\text{px}$. We show the estimated extent and rejection radius for both keypoint types in Fig. 5. Finally, we set the detection score of an accepted keypoint candidate equal to its corresponding score in the associated heatmap and denote it as $s_k(\boldsymbol{y}_k)$.

F. Leaf Instance Segmentation

Next, we perform instance segmentation of sugar beet leaves by employing the predicted offsets $\boldsymbol{R}_\text{leaf}(\boldsymbol{x}_\text{f})$ to assign each

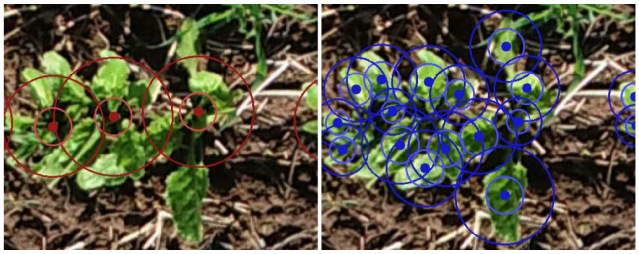

Fig. 5: Left: Predicted stem keypoints with their approximated extent λ_{stem} (outer circle) and rejection radius δ_{stem} (inner circle) for non-maximum suppression. Right: Same illustration for leaves.

foreground pixel $\boldsymbol{x}_\text{f}$ to its nearest, previously detected leaf keypoint. First, we define a metric to compute the distance from any leaf keypoint $\boldsymbol{y}_\text{leaf}$ to a translated foreground pixel $\boldsymbol{x}'_\text{f} = \boldsymbol{x}_\text{f} + \boldsymbol{R}_\text{leaf}(\boldsymbol{x}_\text{f})$ as:

$$D(\boldsymbol{x}'_\text{f}, \boldsymbol{y}_\text{leaf}) = \frac{\|\boldsymbol{x}'_\text{f} - \boldsymbol{y}_\text{leaf}\|_2}{\lambda_\text{leaf}(\boldsymbol{y}_\text{leaf})}, \tag{5}$$

where λ_leaf is the approximated extent in Eq. (4). Thus, this metric is scale-aware and accounts for differently sized leaves that may be spatially close. Next, we assign $\boldsymbol{x}_\text{f}$ to its nearest leaf keypoint $\boldsymbol{y}_\text{leaf}(l)$ in case:

$$D(\boldsymbol{x}'_\text{f}, \boldsymbol{y}_\text{leaf}) \leq \min\left(1.0, \frac{d_\text{max}}{\lambda_\text{leaf}(\boldsymbol{y}_\text{leaf})}\right), \tag{6}$$

but otherwise discard it to reduces false positives. Let d_max be the maximum acceptance distance set to $30\,\text{px}$ that we normalize by the object's extent in accordance with Eq. (5). We determine this hyperparameter based on the validation set. This process assigns a subset of foreground pixels $\mathcal{X}_l \subset \mathcal{X}_f$ to a specific leaf keypoint $\boldsymbol{y}_\text{leaf}(l)$. We define $\mathcal{X}_l$ as the pixel-wise mask of the leaf instance l, as shown in Fig. 1. Subsequently, we follow the work by Papandreou et al. [18] and define the detection score of a leaf instance l as:

$$S_\text{leaf}(l) = \frac{s_\text{leaf}\left(\boldsymbol{y}_\text{leaf}(l)\right)}{|\mathcal{X}_l|} \sum_{\boldsymbol{x}_\text{f} \in \mathcal{X}_l} \exp\left(-D\left(\boldsymbol{x}'_\text{f}, \boldsymbol{y}_\text{leaf}(l)\right)^2\right), \tag{7}$$

which corresponds to the confidence of the leaf keypoint $s_\text{leaf}\left(\boldsymbol{y}_\text{leaf}(l)\right)$ multiplied by the average confidence about the pixels being correctly assigned to this leaf instance. Thus, the score increases in case the translated pixels are spatially closer to their associated keypoint.

G. Plant Instance Segmentation

Finally, we employ the predicted offsets $\boldsymbol{R}_\text{stem}(\boldsymbol{x}_\text{f})$ to group each detected leaf instance to a specific stem keypoint. First, we define a metric to compute the average distance from any stem keypoint $\boldsymbol{y}_\text{stem}$ to subset of translated foreground pixels previously assigned to a leaf instance l as:

$$\bar{D}(l, \boldsymbol{y}_\text{stem}) = \frac{1}{|\mathcal{X}_l|} \sum_{\boldsymbol{x}_\text{f} \in \mathcal{X}_l} \frac{\|\boldsymbol{x}''_\text{f} - \boldsymbol{y}_\text{stem}\|_2}{\lambda_\text{stem}(\boldsymbol{y}_\text{stem})}, \tag{8}$$

where $\boldsymbol{x}''_\text{f} = \boldsymbol{x}_\text{f} + \boldsymbol{R}_\text{stem}(\boldsymbol{x}_\text{f})$. As before, we assign a leaf instance l to its nearest stem keypoint $\boldsymbol{y}_\text{stem}(b)$ in case:

$$\bar{D}(l, \boldsymbol{y}_\text{stem}) \leq \min(1.0, \frac{d_\text{max}}{\lambda_\text{stem}(\boldsymbol{y}_\text{stem})}). \tag{9}$$

This process assigns a set of leaf instances $\mathcal{Y}_b$ to a specific sugar beet instance b. We define the union of all leaves in $\mathcal{Y}_b$ as the pixel-wise mask of this sugar beet instance. However, we discard leaf instances not assigned to any stem keypoint. Finally, we define the detection score of b as:

$$S_\text{beet}(b) = s_\text{stem}(\boldsymbol{y}_\text{stem}(b)) \max_{\boldsymbol{l} \in \mathcal{Y}_b} \exp\left(-\bar{D}\left(l, \boldsymbol{y}_\text{stem}(b)\right)^2\right), \tag{10}$$

which corresponds to the confidence of the stem keypoint multiplied by the score of the most confident leaf associated to this plant. Thus, we consider the presence of at least one leaf associated with a specific stem keypoint as an indicator for the existence of a plant instance.

IV. Experimental Evaluation

The main focus of this work is to perform a joint instance segmentation of sugar beets and their individual leaves based on large-scale orthomosaics captured by UAVs. We perform experiments to support our claims: Our single-stage bottom-up approach (i) detects plant-specific leaf and stem keypoints via predicted offsets, (ii) performs an instance segmentation of crop leaves that we associate to its corresponding plant to conduct a joint instance segmentation of whole sugar beets, and (iii) achieves higher performance w.r.t. Mask-RCNN.

Experimental Setup. We evaluate the performance of our approach in comparison with Mask R-CNN [5] often applied in the agricultural domain [3]. In contrast to our method, it is a top-down approach that detects the bounding box of each instance and subsequently generates its binary mask. However, we train two networks based on Mask-RCNN since a single model cannot perform a joint instance segmentation of plants and leaves. We train the former network to perform an instance segmentation of crop leaves and the latter network to perform instance segmentation of crop plants. Finally, we merge the predictions of both: First, we compute for each predicted leaf the centroid of its mask and subsequently assign it to the plant in whose mask this centroid lies.

Dataset. We evaluate our approach on orthorectified RGB imagery recorded by an UAV that covers a field trial consisting of multiple, spatially separated breeding plots with a ground sampling distance of around $1.5\,\frac{\text{mm}}{\text{px}}$. The size of each breeding plot is about $1.5\,\text{m} \times 8\,\text{m}$ and contains a specific plant breeding experiment. We designed these plots with a small plant spacing and a high weed pressure for challenging conditions. The dataset consists of four spatially separated breeding plots recorded on five different sessions during five weeks. Thus, we obtain for each breeding plot a time series consisting of five orthomosaics. Finally, this results in 20 large-scale orthomosaics of individual breeding plots with plants at different growth stages. We use the entire time series of the first and second breeding plot for training, resulting in 10 orthomosaics. In contrast, we use the time series of the third breeding plot for validation, consequently containing 5 orthomosaics. Finally, we report the results on the remaining five orthomosaics of the time series of the fourth breeding plot. The size of each orthomosaic is about $4320\,\text{px} \times 4100\,\text{px}$. Note that there is no overlap between the train, val, and test

TABLE I: Evaluation of semantic segmentation.

		beet			weed		
Approach	**Date**	P ↑	R ↑	F_1 ↑	P ↑	R ↑	F_1 ↑
Ours	09/02	0.85	**0.90**	**0.88**	0.75	0.82	0.78
Mask R-CNN		**0.94**	0.65	0.77	-	-	-
Ours	09/06	0.80	**0.95**	**0.87**	0.65	0.76	0.70
Mask R-CNN		**0.92**	0.61	0.74	-	-	-
Ours	09/11	0.85	**0.95**	**0.90**	0.67	0.83	0.74
Mask R-CNN		**0.94**	0.66	0.77	-	-	-
Ours	09/20	0.80	**0.95**	**0.87**	0.64	0.80	0.71
Mask R-CNN		**0.95**	0.67	0.78	-	-	-
Ours	10/07	0.93	**0.93**	**0.93**	0.84	0.91	0.88
Mask R-CNN		0.93	0.64	0.76	-	-	-

split since each breeding plot is spatially separated. We provide a analysis together with the published dataset.

Training Details. The original orthorectified images require an impracticable amount of memory on the GPU. Thus, we crop smaller image patches of size $224\,\text{px} \times 224\,\text{px}$ and apply data augmentation via image rotation, scaling, shearing, and flipping. We employ a multi-task loss as the uniformly weighted sum of Eq. (1), Eq. (2), and Eq. (3) to train our network with a batch of six images using Adam [6].

Inference Details. At inference time, we propose a sliding window approach to extract overlapping image patches appropriate for our network. Specifically, we use a window size of $224\,\text{px} \times 224\,\text{px}$ with a stride of $56\,\text{px}$. We emphasize that the outputs of our network allow us to average the predictions in case of overlapping windows. In contrast, the predictions of other approaches [25] do not provide any geometric interpretation that can be averaged in a meaningful way. Thus, these methods cannot be applied to our large-scale imagery. Additionally, we avoid difficult boundary regions by selecting a center crop of $128\,\text{px} \times 128\,\text{px}$ from the network's predictions and fuse only this into the final output that has the same size as the original image. After the aggregation, we perform the keypoint detection and predict the pixel-wise masks of leaf and plant instances.

A. Evaluation Metrics

We evaluate our approach on task-specific metrics that correspond to each prediction of our network.

Semantic Segmentation. To evaluate the semantic segmentation, we compute the precision (P) and recall (R) for the categories beet and weed [11], [15] by comparing the predicted and ground-truth category for each pixel and also provide the F_1 score as their harmonic mean.

Keypoint Detection. We evaluate the keypoint extraction by processing the predicted keypoints in descending order w.r.t. their score. We consider a keypoint as true positive if its distance to a ground-truth keypoint is less than a predefined threshold θ and if the ground-truth was not encountered before. Otherwise, we count it as a false positive. Conversely, we consider all ground-truth keypoints that are not matched to any predicted keypoint as false negatives. We set the threshold θ, derived from the average leaf area, to $20\,\text{px}$ and report the precision, recall, and average precision (AP_θ). We also report the total difference in count (DiC) between the number of predicted and ground-truth keypoints.

Instance Segmentation. We follow a similar procedure to evaluate the quality of pixel-wise instance masks but compute the intersection over union (IoU) for each combination of predicted and ground-truth mask. We consider a predicted instance mask as true positive if the IoU is greater than a predefined threshold and the ground-truth mask is not already assigned to any other prediction. Otherwise, we count it as a false positive. In contrast, we consider all ground-truth masks that cannot be assigned to any predicted mask as false negatives. We set the IoU threshold to 0.5 and report the precision, recall, and average precision (AP50). In addition, we follow [9] and compute the average precision for different IoU thresholds $\in [0.5, 0.95]$ with step size 0.05 and report their average, denoted as AP. Finally, we compute the mean deviation of the leaf count per plant (MADiC).

B. Performance of Our Approach

We report the performance of our approach in comparison with Mask R-CNN based on the evaluation of the test set.

Semantic Segmentation. First, we show that our approach outperforms the baseline w.r.t. the semantic segmentation of sugar beets. This stage is essential since our instance segmentation is based on these results. Thus, any unidentified sugar beet pixel decreases the performance in the following steps. For Mask R-CNN, we assign any pixel of a detected instance to the category beet to derive its semantic segmentation implicitly. We emphasize that we do not report any results for Mask R-CNN w.r.t. weeds since the dataset does not contain weed instance labels required to train these models.

In Tab. I, we show the results for both methods. Our approach achieves superior performance regarding the recall across all sessions and detects more than $90\,\%$ of all pixels for the sugar beet class. However, we perform worse regarding precision except for the last session containing the largest plants, where both methods achieve $93\,\%$. We consider this less problematic since false positive pixels can still be discarded in Eq. (6) during instance segmentation. Otherwise, this can result in less accurate instance masks. However, the F_1 score indicates that we achieve a better trade-off between both metrics. In addition, our method can effectively exploit the semantic information about weeds in the ground-truth labels to segment weed pixels at inference, as describe in Sec. III-C. This provides crucial information about weed coverage.

Keypoint Detection. Next, we evaluate the detection of plant-specific keypoints used to perform the subsequent instance segmentation. For Mask R-CNN, we define a keypoint as the centroid of the predicted instance mask. We show that our approach detects these keypoints more accurately.

First, we analyze the performance w.r.t. leaf keypoints for both methods in Tab. II. Our approach consistently shows a higher recall for sugar beets across all growth stages, e.g., it is always above $88\,\%$, while Mask R-CNN never surpasses $80\,\%$. Consequently, we detect more ground-truth leaf keypoints, as shown in Fig. 6. However, we notice a worse performance regarding the precision as our approach predicts a higher total number of leaf keypoints compared to the ground-truth value denoted by #. This happens more frequently across the intermediate sessions indicated by positive values in the metric DiC and can result in an overcount of leaves. In

TABLE II: Evaluation of keypoint detection.

Approach	Date	Type	#	P ↑	R ↑	AP_θ ↑	DiC → 0
Ours	09/02	leaf	482	**0.92**	**0.91**	**0.88**	**−5**
Mask R-CNN				0.86	0.79	0.77	−37
Ours	09/06	leaf	475	0.89	**0.95**	**0.90**	**30**
Mask R-CNN				**0.92**	0.76	0.74	−83
Ours	09/11	leaf	576	0.92	**0.95**	**0.92**	**15**
Mask R-CNN				**0.95**	0.72	0.72	−136
Ours	09/20	leaf	583	0.90	**0.93**	**0.90**	**25**
Mask R-CNN				**0.93**	0.77	0.75	−104
Ours	10/07	leaf	813	0.92	**0.88**	**0.85**	**−33**
Mask R-CNN				**0.95**	0.80	0.79	−127
Ours	09/02	stem	141	0.89	**0.96**	**0.89**	**11**
Mask R-CNN				**0.97**	0.88	0.86	−13
Ours	09/06	stem	141	0.94	**0.98**	**0.94**	**6**
Mask R-CNN				**0.95**	0.81	0.79	−21
Ours	09/11	stem	156	0.96	**0.97**	**0.96**	**2**
Mask R-CNN				**0.97**	0.85	0.85	−19
Ours	09/20	stem	159	**0.93**	**0.93**	**0.91**	**1**
Mask R-CNN				0.88	0.79	0.74	−16
Ours	10/07	stem	172	**0.94**	**0.92**	**0.90**	**−3**
Mask R-CNN				0.83	0.78	0.71	−10

TABLE III: Evaluation of instance segmentation.

Approach	Date	Type	#	P ↑	R ↑	AP_{50} ↑	AP ↑	MADiC ↓
Ours	09/02	leaf	482	**0.85**	**0.84**	**0.78**	**0.36**	-
Mask R-CNN				0.57	0.53	0.48	0.12	-
Ours	09/06	leaf	475	**0.84**	**0.89**	**0.79**	**0.37**	-
Mask R-CNN				0.74	0.61	0.58	0.20	-
Ours	09/11	leaf	576	**0.85**	**0.87**	**0.82**	**0.39**	-
Mask R-CNN				0.77	0.59	0.56	0.21	-
Ours	09/20	leaf	583	0.83	**0.86**	**0.80**	**0.39**	-
Mask R-CNN				**0.84**	0.69	0.66	0.27	-
Ours	10/07	leaf	813	0.85	**0.82**	**0.76**	**0.44**	-
Mask R-CNN				**0.87**	0.73	0.71	0.32	-
Ours	09/02	plant	141	0.88	**0.94**	**0.89**	**0.56**	**0.32**
Mask R-CNN				**0.95**	0.86	0.84	0.29	0.81
Ours	09/06	plant	141	**0.90**	**0.94**	**0.90**	**0.53**	**0.35**
Mask R-CNN				0.84	0.72	0.68	0.21	0.63
Ours	09/11	plant	156	**0.95**	**0.96**	**0.94**	**0.53**	**0.42**
Mask R-CNN				0.86	0.76	0.71	0.23	0.85
Ours	09/20	plant	159	**0.87**	**0.87**	**0.83**	**0.44**	**0.56**
Mask R-CNN				0.82	0.74	0.71	0.24	0.93
Ours	10/07	plant	172	**0.88**	**0.85**	**0.82**	**0.42**	**0.79**
Mask R-CNN				0.75	0.70	0.64	0.21	0.99

contrast, Mask R-CNN underestimates the total number of leaf keypoints by a large margin of up to 136. Consequently, Mask R-CNN achieves a higher precision but performs worse regarding the recall. Still, we achieve a better trade-off between precision and recall stated by higher values w.r.t. AP_θ with above 0.85 across all sessions, see Tab. II.

Next, we analyze the performance w.r.t. stem keypoints for both methods in Tab. II. For our approach, we observe a similar behavior as before. We consistently achieve a recall above 92 % across all sessions, while Mask R-CNN does not surpass a recall of 88 %. However, we perform worse regarding precision in the first three sessions, containing sugar beets of early growth stages. This can result in false positive instances biasing the estimated number of plants. In contrast, we outperform Mask R-CNN in this metric for later growth stages by up to 11 percent points in the last session. Again, Mask R-CNN constantly underestimates the total number of stem keypoints by up to 21. This results in an increased precision at the expense of recall. However, we still achieve an increased trade-off between precision and recall as reported by higher values w.r.t. AP_θ, see Tab. II.

Instance Segmentation. The last experiments show the performance of the instance segmentation. The results support our claim that our approach provides pixel-wise masks for crop leaves and plants and also associates each leaf effectively to a specific plant, see Fig. 6. We show that our approach outperforms the baseline regarding this task.

First, we provide quantitative results w.r.t. the instance segmentation of leaves in Tab. III. We achieve a consistently high performance regarding precision and recall across all sessions. For Mask R-CNN, we observe a considerable loss in performance at early growth stages. In contrast, we achieve the best performance at intermediate growth stages resulting in a recall of 87 % and a precision of 85 % compared to 59 % or 77 % for Mask R-CNN. We attribute this to leaves that are large enough to be well-detectable without too much mutual overlap. Besides, detecting small objects is a well-known issue [5]. This is supported in terms of AP50, where we achieve consistently scores above 0.76 while Mask R-CNN performs worst (0.48) at the earliest growth stage and best (0.71) at the latest session. Thus, our approach provides more accurate leaf masks for crops at a variety of growth stages.

Finally, we provide quantitative results w.r.t. the instance segmentation of sugar beet plants in Tab. III. We achieve the highest score of 0.94 in terms of AP50 at intermediate growth stages compared to 0.71 by Mask R-CNN, where crops have a considerable size and are well separated. However, the performance decreases for both methods with increasing growth stages since different sugar beets overlap, see Fig. 6. However, we still achieve higher values in such cases. In contrast to the leaf instance segmentation, we note a decrease of performance w.r.t. to the AP with increasing growth stages for our method. We attribute this to crowded in-field conditions with less spacing between plants which makes the task to associate each leaf to a specific plant more difficult. Thus, wrong associations result in less accurate plant instance masks.

For field monitoring, we consider the MADiC particularly important since it evaluates the leaf count per plant that is highly correlated to its growth stage [8]. We achieve superior performance across all sessions, indicating that our method to associate each leaf to a particular plant is more accurate. Generally, this task is less complex at early growth stages since each plant consists of small leaves but increases complexity as plants grow. This is reflected by a small value of 0.32 at the earliest and an increased deviation of 0.79 in the last session.

V. Conclusion

We presented a novel approach that provides a pixel-wise instance segmentation of crops and their associated leaves in large-scale orthorectified imagery. Our vision-based method supports plant breeders and scientists to assess relevant per-plant parameters relevant for automatica field monitoring. We propose CNN that relies on a simple yet effective topological model of plants that associates each pixel to plant-specific keypoints via offsets. Our thorough experimental evaluation using real-world imagery from breeding plots shows that our method performs an accurate instance segmentation compared to state-of-the-art instance segmentation approaches and is well-suited to monitor individual plants at different growth

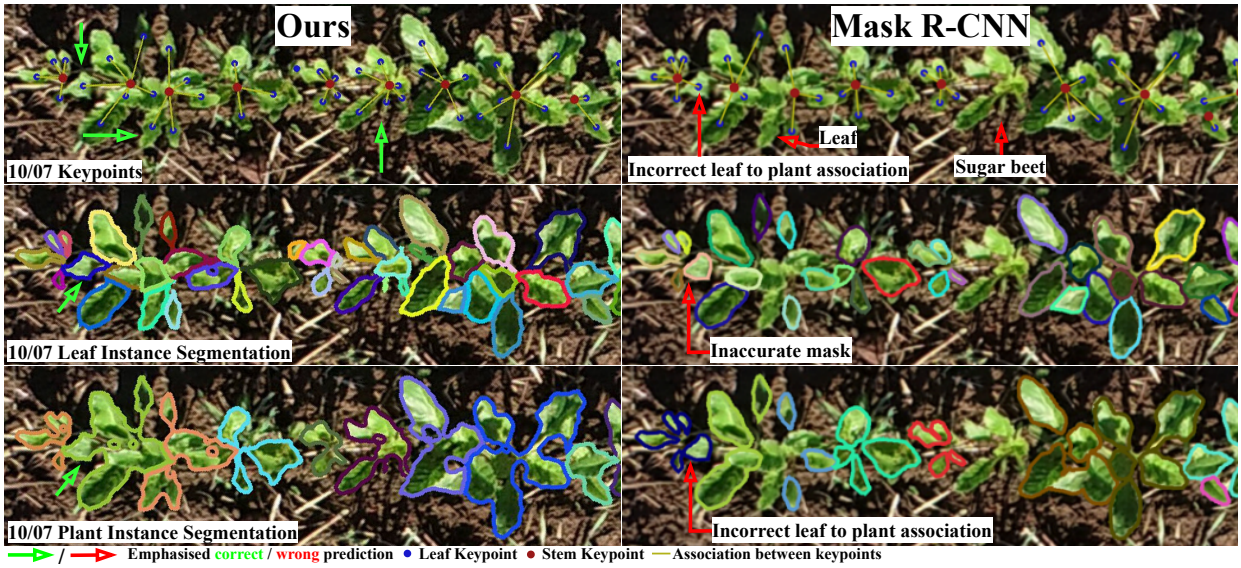

Fig. 6: Qualitative results of leaf and stem keypoints, leaf instance segmentation, and plant instance segmentation of our approach (left) and Mask R-CNN (right) at a particular growth stage. *We show the outline of each instance's pixel-wise mask for reasons of clarity.*

stages. We emphasize that the predictions of our approach are relevant for autonomous ground vehicles to identify the growth stage of individual plants for targeted management actions.

REFERENCES

[1] Agisoft Metashape. Version 1.6.2. Metashape professional edition. 2

[2] B.D. Brabandere, D. Neven, and L.V. Gool. Semantic instance segmentation with a discriminative loss function. In *Deep Learning for Robotic Vision workshop, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[3] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly. Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots. *Applications in Plant Sciences*, 8(7):e11373, 2020. 1, 2, 5

[4] R.T. Furbank and M. Tester. Phenomics–technologies to relieve the phenotyping bottleneck. *Trends in Plant Science*, 16(12):635ff, 2011. 1

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017. 2, 5, 7

[6] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2016. 6

[7] V. Kulikov and V. Lempitsky. Instance Segmentation of Biological Images using Harmonic Embeddings. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[8] P. Lancashire, H. Bleiholder, T. Boom, P. Langelüddeke, R. Stauss, E. Weber, and A. Witzenberger. A Uniform Decimal Code for Growth Stages of Crops and Weeds. *Annals of Applied Biology*, 119(3):561–601, 1991. 1, 7

[9] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–755, 2014. 6

[10] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018. 1, 2

[11] P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 3:3097–3104, 2018. 6

[12] P. Lottes, M. Höferlin, S. Sander, M. Müter, P. Schulze-Lammers, and C. Stachniss. An Effective Classification System for Separating Sugar Beets and Weeds for Precision Farming Applications. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016. 2

[13] F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-Based 4D Registration of Plants Point Clouds for Phenotyping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020. 2

[14] A. Milioto, P. Lottes, and C. Stachniss. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017. 1

[15] A. Milioto, P. Lottes, and C. Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 6

[16] A. Milioto and C. Stachniss. Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019. 2

[17] M. Oghaz, M. Razaak, H. Kerdegari, V. Argyriou, and P. Remagnino. Scene and environment monitoring using aerial imagery and deep learning. In *Proc. of the IEEE Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2019. 2

[18] G. Papandreou, T. Zhu, L. Chen, S. Gidaris, J. Tompson, and K. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018. 2, 4, 5

[19] C. Potena, R. Khanna, J. Nieto, R. Siegwart, D. Nardi, and A. Pretto. AgriColMap: Aerial-ground collaborative 3D mapping for precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1085–1092, 2019. 2

[20] S. Ren, K. He, R. Girshick, and J. .Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[21] O. Ronneberger, P.Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *LNCS*, pages 234–241, 2015. 3

[22] H. Scharr, M. Minervini, A. French, C. Klukas, D. Kramer, X. Liu, I. Luengo, J. Pape, G. Polder, and D. Vukadinovic. Leaf Segmentation in Plant Phenotyping: a Collation Study. *Machine Vision and Applications*, 27(4):585–606, 2016. 1

[23] W. Shi, R. van de Zedde, H. Jiang, and G. Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019. 2

[24] C. Stachniss, O. Martínez-Mozos, A. Rottmann, and W. Burgard. Semantic Labeling of Places. In *Proc. of the Intl. Symposium on Robotic Research (ISRR)*, San Francisco, CA, USA, 2005. 2

[25] J. Weyler, A. Milioto, T. Falck, J. Behley, and C. Stachniss. Joint Plant Instance Detection and Leaf Count Estimation for In-Field Plant Phenotyping. *IEEE Robotics and Automation Letters (RA-L)*, 6:3599–3606, 2021. 2, 6

[26] Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2018. 3