 DOI: 10.3217/978-3-85125-954-4-06

One-Pixel Instance Segmentation of Leaves

Julia Strebl*, Eric Stumpe*, Thomas Baumhauer, Lena Kernstock, Markus Seidl, Matthias Zeppelzauer
Institute of Creative Media Technologies, St. Pölten University of Applied Sciences
{firstname}.{lastname}@fhstp.ac.at

Abstract

The segmentation of plant leaves is an essential prerequisite for vision-based automated plant phenotyping applications like stress detection, measuring plant growth and detecting pests. Segmenting plant leaves is challenging due to occlusions, self-shadows, varying leaf shapes, poses and sizes and the presence of particularly fine structures. We present a novel leaf segmentation approach that takes single pixels as input to initialize the segmentation of leaves. Additionally, we introduce a new strategy for transfer learning that we call "tandem learning" which enables the integration of previously learned network representations into a structurally different network. We evaluate different configurations of our approach on publicly available data sets and show that it yields competitive segmentation results compared to more complex segmentation approaches.

1. Introduction

Plant phenotyping refers to methodologies for the characterization of plants, i.e., plant architecture and composition at different scales [4]. This includes the visual assessment of plant traits to investigate plant growth, plant state and plant stress [11]. The manual assessment of these properties from visual observation is an expensive and tedious process. Phenotyping at larger scales thus requires automated methods for the quantification of plant traits. Computer vision approaches can solve plant phenotyping problems at large scales in a non-invasive manner. Thereby, automated leaf segmentation is an essential prerequisite for many downstream tasks including leaf counting, leaf/plant tracking and the detection of plant stress, diseases and pests.

Leaf segmentation is an instance segmentation problem [7], where the goal is to pixel-accurately segment objects of the same type (here leaves). Plants pose a number of challenges to this task including (i) coping with complex background (e.g., from soil visible in the images, trunks, branches etc.); (ii) handling fine structures (e.g., the stems of the leaves); (iii) solving occlusion problems introduced by overlapping leaves; (iv) coping with differently sized and shaped leaves (e.g., due to different ages) and different leaf poses; and (v) handling shadowing and varying reflectivity of differently oriented leaves [19].

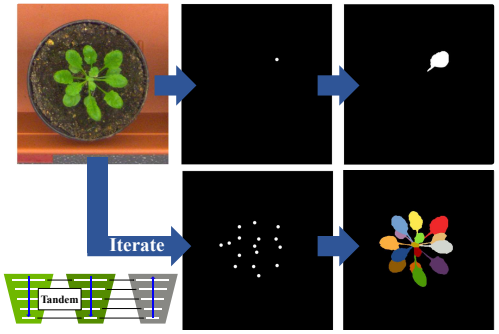

Figure 1. One-pixel instance segmentation: our approach first learns to estimate useful seed points for leaf segmentation and then segments leaves from these seed points via *tandem learning*, a more flexible form of traditional transfer learning.

In this paper, we present a simple and thus robust leaf segmentation approach that achieves promising segmentation results on established benchmark data sets. Our approach is anchor-free and thus makes no *a priori* assumptions about leaf size and shapeand can principally learn arbitrary leaf shapes. In our approach we introduce two novel concepts for instance segmentation (see also Figure 1):

- *One-pixel segmentation*: a form of instance segmentation that requires only minimal input, i.e., a single seed pixel to segment an object instance. One-pixel segmentation makes our approach equally suitable for fully automated and interactive segmentation, which is usually hard for fully end-to-end trained methods.
- *Tandem learning*: a new form of transfer learning that helps to incorporate existing knowledge captured in a pre-trained network in a novel task that requires a *structurally different* network architecture.

We design and evaluate different configurations of our ap-

*both authors contributed equally to this paper

proach and perform ablation studies to evaluate the influence of the individual processing steps.

2. Related Work

Segmentation methods can be split into anchor-based and anchor-free approaches. Here, we review both types to place our approach in context. We further review related methods that inspired our approach.

Anchor-Based Instance Segmentation. A common strategy for instance segmentation is the utilization of predefined anchor boxes for generating region proposals. A popular network of this category is Mask-RCNN [8]. In Mask-RCNN, first image features are extracted, followed by the prediction of object classes and Regions of Interest (RoIs), which is facilitated by the initial anchor boxes. In a second step, segmentation masks are predicted from the proposed RoIs. Huang et al. [9] introduced a separate Intersection over Union (IoU) prediction branch to Mask-RCNN to increase performance. Liu et al. [13] further improved the architecture by using a bottom-up path augmentation scheme for the extraction of image features. Other follow-up works focus on aspects such as inference speed [2] or object border refinement [10]. To get optimal results for different types of image data sets, preset anchor boxes and their dimensions have to be adapted to the dimensions of the target objects. Since our method does not require anchors it is not subject to this restriction.

Anchor-Free Instance Segmentation. Tian et al. [22] demonstrated an effective method for object detection that does not require the use of anchor boxes. Instead, distances to the nearest bounding box and its dimensions are directly learned and represented as a 4D feature map. This work inspired other authors to adopt this method for region proposal-based instance segmentation. Bounding box-based methods in general work best for objects with similar height and weight, but can fail for elongated objects that overlap as demonstrated in [3]. Consequently a strand of research has evolved using different working principles to avoid this issue. Bai and Urtasun [1] predict the per-pixel angle to the nearest object border, enabling the segmentation of instances through their computed watershed energy level. De Brabandere et al. [3] formulate instance segmentation as a per-pixel problem, where the discriminative loss function enforces pixels of the same object to be close in latent space. Our work falls into the group of anchor-free instance segmentation methods and uses automatically estimated seed points in combination with a trained instance model to iteratively segment leaves.

Leaf Instance Segmentation. Gomes and Zheng [5] adopted a standard Mask-RCNN architecture for leaf segmentation and demonstrated that leaf masks of high quality can be predicted by employing simpler strategies, such as threshold adjustment and test time augmentations. To simulate the counting process of humans, Ren and Zemel [16] utilized a recurrent neuronal network (RNN), which sequentially proposes new regions of interest based on an attention mechanism. Guo et al. [6] devised a multi-scale attention module and mask refining module to improve the segmentation quality of their instance segmentation model. Wolny et al. [24] introduced a technique, which can also deal with sparsely labelled instance annotations and is based on the pixel embedding method in [3]. Feeding perturbations of the same input image to two embedding networks, a penalty is applied if both predicted masks are not geometrically consistent, thus enforcing constraints for the embedding space leading to better segmentation accuracy. In contrast to existing methods, our network architecture is more simple and straightforward and works well with already established loss functions such as binary cross-entropy.

Interactive Instance Segmentation. We further draw inspiration from interactive segmentation approaches. In recent methods, users can draw positive and negative object regions to guide the segmentation process [25], or are involved in a human-in-the-loop process where they actively annotate pixels of regions which are difficult to segment [20]. Lin et al. [12] developed an approach, in which interactive segmentation is guided by multiple user clicks with a focus on the first click acting as a segmentation anchor. Our goal for the future is to advance our method for efficient and low-effort interactive segmentation, which is facilitated by our one-pixel segmentation strategy.

3. Approach

An overview and illustration of our approach is shown in Figure 2. Below, we describe the individual steps in detail.

3.1. Data Preparation

Input data for our approach are RGB images of plants (see also Section 4). Additionally, for foreground segmentation we use binary segmentation masks as ground truth. For instance segmentation, we use masks including individual leaf annotations (multi-labeled ground truth masks). The training of the leaf instance segmentation model further requires the computation of masks, which specify the center for each leaf. In these masks the center pixel is highlighted by a value of 1, while all other values are 0. These masks can easily be created from the multi-labeled ground truth masks by applying e.g., distance transform and peak detection on each instance's area.

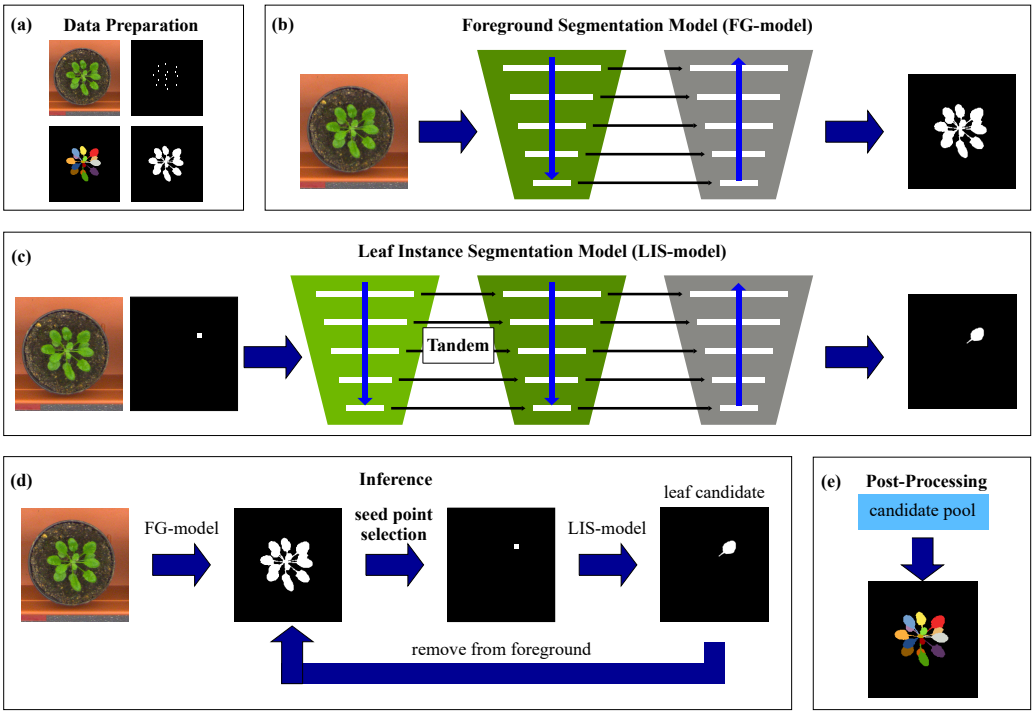

Figure 2. Overview of our leaf instance segmentation approach. First, we identify relevant image regions corresponding to leaves of the plant via semantic segmentation, see "FG-model" in (b). The result is a binary segmentation that captures the entire leaf tissue. From this segmentation we estimate potential leaf centers, which serve as seed points for *one-pixel instance segmentation*. The seed points are added as additional input channel to the leaf instance segmentation model, see "LIS-model" in (c). Using the proposed *tandem learning* scheme, a pre-trained encoder is incorporated into the LIS-model to accelerate training. The LIS-model segments one leaf at a time and is iteratively called to successively segment all leafs of the plant (d). Post-processing (e) consolidates the individual instance segments.

3.2. Training

3.2.1 Foreground Segmentation

For foreground segmentation we employ an encoder-decoder architecture with skip connections, similar to U-Net [17]. A pre-trained VGG16 backbone [21] serves as encoder [18]. The architecture of the decoder mirrors that of the VGG backbone, but instead of max-pooling layers we use up-convolutional layers (4 layers) to bring the feature maps back to the input image dimensions. In addition, the decoder receives feature maps through skip connections which are thereby incorporated in the training process. We use RGB images as input, binary segmentation masks as learning target, and binary cross-entropy as loss function.

3.2.2 Leaf Instance Segmentation

The LIS-model is also based on the U-Net architecture [17] from Section 3.2.1, but has two encoders A and B (see Figure 3) which are connected side-by-side in a tandem. Both encoders compute feature maps at different scales, which are concatenated with each other along the depth dimension. This architecture, which we call a "*tandem architecture*" enables to combine network models (here two encoders) designed for different types of inputs.

As Encoder A we use VGG16 [21], which has been fine-tuned during foreground segmentation and takes three-channel RGB images as input. Therefore, the network is already capable of extracting meaningful plant-related features from RGB images. Encoder B receives images with a channel size of 4: the RGB channels plus the center point mask of a given leaf instance. Since no pre-trained model exists for this type of input, the model is initialized with random weights. Encoder B is further connected to the decoder in the same fashion as in the U-Net architecture [17]. All layers of the tandem network are fine-tuned/trained.

The tandem architecture should foster the integration of previously learned knowledge into a new learning task, which requires a different input (and potentially output) structure. This architecture is more flexible than standard transfer learning where usually the input is required to be equivalent and only the output layer is adapted. Additionally, it enables to combine two simple network architectures (VGG and U-Net) avoiding the need for a more complex (and more difficult to train) architecture.

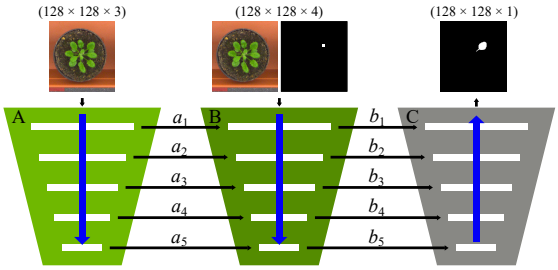

Figure 3. The concept of *tandem learning*: two encoders A and B are connected side by side. Thereby, A and B may have different input structure. Via connections a_i pre-learned information from A is shared with B. The final output is generated by decoder C.

Essential for training the network is data augmentation. Aside from conventional image transforms (see Section 4.3), we adjust the fourth input channel to make the network less dependent on the actual leaf center location. We propose two augmentation methods. First, instead of taking the leaf mask with the exact leaf center, a random pixel from the area of the leaf is taken. Second, starting from the exact center we specify a radius r that is increased by one pixel with each epoch. For augmentation, pixels are chosen at random that lie within this increasing radius. This facilitates location invariance in the optimization.

3.3. Inference

The goal of inference is to utilize both trained models in a combined manner to segment all leaves in an input image in absence of ground-truth. First, the foreground mask of the whole plant is computed with the FG-model. From this segmentation, we estimate potential center points automatically to initiate leaf instance segmentation. To select appropriate seed candidates, we propose two methods:

Distance transform (DT) selection (sorted/unsorted): First, morphological erosion is applied to the foreground mask to separate leaves that are loosely connected (i.e., touching each other). Next, the DT is computed for each connected region. The seed candidate is then selected at the location of the maximum value of the DT. Optionally, we sort the connected regions by area to start segmentation with the largest potential leaf.

Gaussian kernel selection (sorted/unsorted): The 2D convolution of the foreground segmentation with a Gaussian 2D kernel is computed. In the result image, pixels close to leaf borders have low values, since foreground (value 1) and background pixels (value 0) are in the effective range of the Gaussian kernel. Pixels in the center of leaves, however, yield high output values (only foreground in the effective range). We apply 2D peak detection to identify potential leaf centers. The 2D Gaussian kernel has 15×15 pixels and a sigma of 7. As in the first method, we optionally sort the connected regions by area.

Following the selection of seed candidates, the trained LIS-model is used to predict the leaf instance mask. Next, the segmented leaf is added to a pool of leaf candidates and the mask of this leaf is subtracted from the foreground mask. This assures that no seed candidates are selected in an already segmented area, which would lead to repeated segmentation of the same leaf. Inference repeats and keeps adding new leaf instances to the pool of leaf candidates until the foreground segmentation mask is empty.

3.4. Post-Processing

The result of leaf instance segmentation is a set of potentially overlapping leaf candidate regions. Noisy foreground segmentation may lead to oversegmentation (too many leaf candidates). Post-processing aims to compensate this by fusing only partially segmented leaves. We propose three strategies for consolidating leaf segments: (i) *deleting*, (ii) *merging* and (iii) *intersecting*. Thereby, all leaf candidate regions are compared via Intersection over Union (IoU) to estimate their mutual overlap. IoU is used as criterion to decide how to proceed with the two candidates as follows:

- Strategy *deleting* is based on the hypothesis that our leaf segmentation model performs better on large leaves. As soon as the IoU threshold for two candidate segments is exceeded, the smaller one is deleted.
- In *merging* two overlapping segments are joined together when their IoU is in a certain range. Hereby, we account for only partially detected leaves, i.e., cases where one leaf is over-segmented.
- In strategy *intersecting* only those leaf areas are preserved, which are supported by more than one candidate segment. This should help to increase the robustness of the segmentation.

The two latter methods facilitate the merging of leaves with a significant overlap and at the same time avoid that adjacent and touching leaves are merged.

4. Experimental Setup

4.1. Datasets

We employ publicly available data sets to facilitate performance comparisons with other methods. The first data set is subset "A1" from the *Plant Phenotyping Dataset (PPD)* introduced in [14, 15], which consists of 128 manually annotated images. To show how well our approach generalizes to other types of data and plants, we further evaluate our

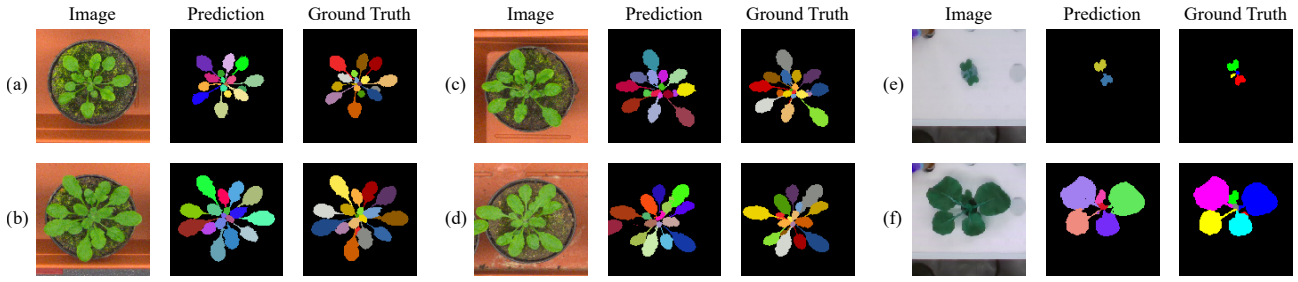

Figure 4. Instance segmentation results from our method for test images of the Plant Phenotyping Dataset (a-d) and KOMATSUNA (e,f).

method on the KOMATSUNA data set [23], that comprises 300 semi-automatically annotated images. Data has been split into 80% training and 20% testing for all experiments.

4.2. Performance metrics

To assess training progress for both models in our approach we utilize Intersection over Union (IoU) and Dice similarity coefficient (DSC). As proposed by [14,15] the final instance segmentation results are measured with the Symmetric Best Dice (SBD) measure, which is particularly designed for instance segmentation problems and can cope with different but equivalent label assignments. All metrics in our experiments are averaged over three complete repetitions with different random initializations of the network weights.

4.3. Parameters

Our approach has a number of hyperparameters and configuration options, which we evaluate in this paper. For object center estimation we evaluate both strategies from Section 3.3 with sorted and unsorted components. For post-processing we evaluate the three strategies from Section 3.4). For strategy deleting we apply an IoU threshold of 0.7, for merging an IoU range between 0.1 and 0.5 and for intersecting an IoU threshold of 0.5 (suitable parameter values were found via grid search in a preliminary experiment). The training parameters for the foreground segmentation network are as follows: training is conducted for 40 epochs with a learning rate of 0.00001 and batch size 20.

Downscaled RGB images of size $128 \times 128 \times 3$ serve as the network input. For the LIS-model, training (input size $128 \times 128 \times 4$) is initiated for 150 epochs with a batch size of 32 and a learning rate of 0.0001. For both models, binary cross-entropy and Adam optimizer are applied. To aid the learning process, we employ random geometrical (flipping, zooming, shifting, rotating, shearing) and color data augmentation (noise, brightness, contrast) in addition to the augmentation of leaf centers as described in Section 3.2.2.

5. Results

Overall performance. The overall instance segmentation performance of our approach in terms of SBD is shown in Table 1. Additionally, we provide Dice and IoU for foreground segmentation and leaf instance segmentation. The highest scores for the PPD are achieved with center estimation via distance transform selection (no sorting) and post-processing via deleting strategy. Similarly, the highest scores for KOMATSUNA are achieved with distance transform selection (sorted) and deleting strategy. However, also Gaussian kernel selection and intersection strategy lead to the same peak performance, showing that the robustness of center estimation and post-processing strategy is high.

Tandem training. To evaluate the tandem architecture for transfer learning we perform an ablation experiment by removing the second encoder in the LIS model. The result is an average performance drop of 2.1% in SBD for the PPD and 1.8% for KOMATSUNA. We notice during our experiments that the training in tandem fashion leads to a faster and smoother convergence of the training loss compared to training without tandem. This shows that tandem learning is a suitable approach to take benefit of a previously learned representation, even if it has a different input structure.

Instance center estimation. Here, we evaluate the different leaf center estimation strategies from Section 3.3 systematically and the sensitivity of results to different choices. Results (see Table 2) show that Distance transform selection provides the highest performance across both data sets.

Post-processing strategies. Similarly, as above, we evaluate the different post-processing strategies introduced in Section 3.4. Table 2 shows their impact on overall results. We conclude that delete and intersection outperform post-processing via merging throughout all experiments.

Performance comparison. To objectively assess our results, we compare them with state-of-the-art results from the literature for both data sets, see Table 3 for a listing. For the PPD we achieve comparable scores to both De Brabandere et al. [3] and Ren and Zemel [16] and outperform the approaches reported in [19]. The most recent approaches still outperform our results, which may be due to the higher complexity of the approaches. An additional factor might

Data set	FG IoU	FG Dice	LIS IoU	LIS Dice	SBD
Plant Phenotyping	0.862 (±0.0014)	0.928 (±0.010)	0.819 (±0.012)	0.882 (±0.011)	0.832 (±0.008)
KOMATSUNA	0.871 (±0.006)	0.930 (±0.003)	0.754 (±0.033)	0.836 (±0.030)	0.754 (±0.005)

Table 1. Overall segmentation results of our approach for both evaluated data sets.

	Plant Phenotyping Dataset A1			KOMATSUNA		
	delete	merge	intersection	delete	merge	intersection
DTS unsorted	**0.831** (±0.0032)	0.825 (±0.0020)	**0.831** (±0.0036)	0.719 (±0.0155)	0.710 (±0.0193)	0.712 (±0.0169)
DTS sorted	0.808 (±0.0037)	0.807 (±0.0028)	0.807 (±0.0034)	0.747 (±0.0186)	0.738 (±0.0189)	0.750 (±0.0184)
GKS unsorted	0.787 (±0.0029)	0.786 (±0.0003)	0.789 (±0.0029)	0.751 (±0.0110)	0.739 (±0.0095)	**0.754** (±0.0119)
GKS sorted	0.775 (±0.0016)	0.773 (±0.0003)	0.775 (±0.0006)	0.742 (±0.0118)	0.734 (±0.0120)	0.744 (±0.0116)

Table 2. Systematic comparison results for different center estimation strategies (distance transform selection (DTS) sorted/unsorted, Gaussian kernel selection (GKS) sorted/unsorted) and post-processing strategies (delete, merge, intersection).

Method	PPD A1	KOMATSUNA
Scharr et al. [19] (IPK)	0.744	
Scharr et al. [19] (Nottingham)	0.683	
Scharr et al. [19] (MSU)	0.667	
Scharr et al. [19] (Wageningen)	0.711	
De Brabandere et al. [3]	0.849	
Ren and Zemel [16]	0.842	
Gomes and Zheng [5]	0.920	0.745
Guo et al. [6]	0.925	
Wolny et al. [24]	0.920	

Table 3. SBD scores of different methods on A1 subset of the Plant Phenotyping Dataset (PPD) and the KOMATSUNA Dataset.

be that the reported results stem from the leader board[1] and are not 100% comparable as we test our approach on a 20% subset of the training set, while the performance in the leader board refers to a separate test set (for which no labels were available for our experiments). For the KOMATSUNA data set we could identify only one approach [3] for comparison in the literature (see Table 3). The performance obtained by our approach with an SBD of 0.754 slightly outperforms the previously reported result of 0.745.

Qualitative results In Figure 4, exemplary segmentation results for the test sets of the Plant Phenotyping Dataset (a-d) and KOMATSUNA (e,f) are shown. Overall, most separate leaves are segmented accurately and leaf edges are very closely aligned to the ground truth. In (c) and (g) it can be seen that some very small leaves in the center are not correctly segmented. Sometimes also leaves, which are largely covered by other leaves are not segmented well (see leaves in the lower area of (b) and (d)). Examples in (e) and (f) show that leaves with different size and shapes can be segmented well. Remarkable is further that in (d) a leaf of an neighboring plant is correctly segmented, although it is not part of the annotated ground truth.

Limitations Our approach works slightly better for larger objects than for small ones. The reason is that large objects generate more (overlapping) segment candidates, which can be better consolidated and refined via post-processing. Our one-pixel segmentation approach functions well for the segmentation of instances that consist of a single connected region, but can fail for instances that are fragmented (e.g., a leaf that is intersected by the petiole of another leaf, and thus consists of two separate regions).

6. Conclusion

We have presented a novel approach for leaf instance segmentation which uses individual pixels indicating object centers as seed points for instance segmentation. Our approach yields promising results on public benchmark data sets and can compete with much more complex segmentation approaches from literature. Since our approach makes no *a priori* assumptions about the structure, shape and pose of plant leaves, it may be applicable to other instance segmentation tasks and thus may be of broader interest to the community. The same applies to the *tandem training* that we use for transfer learning. Future work will focus (i) on predicting leaf centers during foreground segmentation to replace leaf center estimation during inference and (ii) on demonstrating the broader applicability of the proposed approach for other instance segmentation tasks.

Acknowledgments

This work was funded by the research promotion agency of the province of Lower Austria (GFF), proj. no. FTI18-005.

[1] https://competitions.codalab.org/competitions/18405#results

References

[1] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[3] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

[4] Fabio Fiorani, Ulrich Schurr, et al. Future scenarios for plant phenotyping. *Annu. Rev. Plant Biol*, 64(1):267–291, 2013.

[5] Douglas Pinto Sampaio Gomes and Lihong Zheng. Leaf segmentation and counting with deep learning: on model certainty, test-time augmentation, trade-offs. *arXiv preprint arXiv:2012.11486*, 2020.

[6] Ruohao Guo, Liao Qu, Dantong Niu, Zhenbo Li, and Jun Yue. Leafmask: Towards greater accuracy on leaf segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1249–1258, 2021.

[7] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[9] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6402–6411, Los Alamitos, CA, USA, 2019. IEEE Computer Society.

[10] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[11] Zhenbo Li, Ruohao Guo, Meng Li, Yaru Chen, and Guangyao Li. A review of computer vision technologies for plant phenotyping. *Computers and Electronics in Agriculture*, 176:105672, 2020.

[12] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13339–13348, 2020.

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[14] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A. Tsaftaris. Plant phenotyping datasets. `http://www.plant-phenotyping.org/datasets`, 2015.

[15] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition letters*, 81:80–89, 2016.

[16] Mengye Ren and Richard S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[19] Hanno Scharr, Massimo Minervini, Andrew P French, Christian Klukas, David M Kramer, Xiaoming Liu, Imanol Luengo, Jean-Michel Pape, Gerrit Polder, Danijela Vukadinovic, et al. Leaf segmentation in plant phenotyping: a collation study. *Machine vision and applications*, 27(4):585–606, 2016.

[20] Gyungin Shin, Weidi Xie, and Samuel Albanie. All you need are a few pixels: semantic segmentation with pixelpick. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1687–1697, 2021.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[23] Hideaki Uchiyama, Shunsuke Sakurai, Masashi Mishima, Daisaku Arita, Takashi Okayasu, Atsushi Shimada, and Rinichiro Taniguchi. An easy-to-setup 3d phenotyping platform for komatsuna dataset. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[24] Adrian Wolny, Qin Yu, Constantin Pape, and Anna Kreshuk. Sparse object-level supervision for instance segmentation with pixel embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4402–4411, 2022.

[25] Matthias Zeppelzauer, Georg Poier, Markus Seidl, Christian Reinbacher, Samuel Schulter, Christian Breiteneder, and Horst Bischof. Interactive 3d segmentation of rock-art by enhanced depth maps and gradient preserving regularization. *Journal on Computing and Cultural Heritage (JOCCH)*, 9(4):1–30, 2016.