Analysis of Arabidopsis Root Images — Studies on CNNs and Skeleton-Based Root Topology

Birgit Möller, Berit Schreck, and Stefan Posch

Institute of Computer Science, Martin Luther University Halle-Wittenberg
Von-Seckendorff-Platz 1, 06099 Halle (Saale), Germany

birgit.moeller@informatik.uni-halle.de

Abstract

Roots and their temporal development play an important role in plant research. Over the decades image-based monitoring of root growth has become a key methodology in this research field. The growing amount of image data is often tackled with automatic image analysis approaches. In particular convolutional neural networks (CNNs) recently gained increasing interest for root segmentation. This segmentation of roots is usually only the first step of an analysis pipeline and needs to be supplemented by topological reconstruction of the complete root system architecture.

In this paper we present a comprehensive study of different CNN architectures, loss functions and parameter settings for root image segmentation. In addition, we show how main and lateral roots can be identified based on the skeletons of segmented root components as a first step towards topological reconstruction of root system architecture. We present quantitative and qualitative results on data released in the course of the CVPPA Arabidopsis Root Segmentation Challenge 2021.

1. Introduction

Roots are an important organ of plants. They play essential roles in ensuring secure anchorage and in uptaking vital nutrients from the soil. Hence, root physiology and status have a major impact on growth and development of plants and render them a key topic in plant research. Quantitative data about root mass and root system development over time provide a solid basis for understanding functional relationships between environmental conditions, plant development and the status of ecosystems as a whole [3]. As roots are the plant organs least accessible, in the beginning of root studies direct manual measurements from excavated roots dominated the collection of quantitative data.

Meanwhile image-based techniques are well established. For experiments in soil minirhizotrons are available which allow for non-destructive acquisition of time series data of roots as basis for developmental studies [9]. Such image data does not provide enough information to reconstruct the architecture of the complete root system, but data is restricted to overall root length or mass in the sample area. As an alternative, growing plants in culture medium within transparent plates and regular acquisition of images from such plates (Fig. 1) has become a popular protocol for studying the development of complete root systems [1].

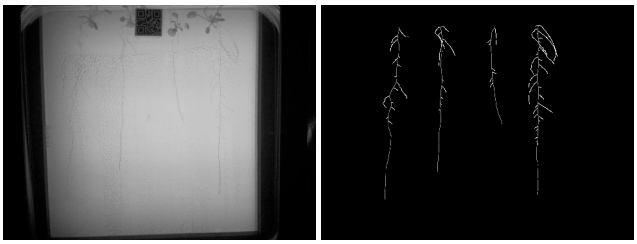

Figure 1. Left, prototypical root image from one of the challenge data sets, right, corresponding ground truth annotation with background in black, lateral roots in white and main roots in gray.

Automatic analysis of such data typically subsumes two analysis stages. In the first stage images are segmented to separate roots from the image background on the pixel level. Essentially this allows to estimate root mass. Root topology is extracted in the second stage where different levels are considered. Identification of the center pixels of the roots allow for length measurements. The reconstruction of complete root system architectures subsumes in addition explicit annotation of branching points, as well as distinction between the main root (MR) and lateral roots.

Over the decades the methodology for root image analysis has emerged from pure manual image annotation to fully automatic approaches. During the last years particularly deep learning with convolutional neural networks (CNNs) has proven suitable for solving the root image segmentation task. As all methods based on deep learning, such ap-

proaches strongly rely on a sufficiently large set of annotated training data which often forms a serious bottleneck.

Two data sets of annotated root images were released for the Arabidopsis Root Segmentation Challenge organized in conjunction with the 7th Workshop on Computer Vision in Plant Phenotyping and Agriculture held as part of the ICCV 2021 (for details see Sec. 4). The task of the challenge is to segment all roots from a given image and to identify the MR and all lateral roots of individual plants (Fig. 1).

In this paper we present a fully automatic approach for solving the challenge task. Root segmentation is performed by applying CNNs to individual images of a time series. We decided to treat the images of a time series independently, as for the challenge training data annotations are available only for few, non-consecutive images of each series. To subsequently identify the MRs in the segmentation results we rely on topological analysis of root skeletons extracted from the segmentation results. Our CNN model with which we successfully participated in the challenge comprises a U-Net architecture with VGG16 backbone and was trained by first applying a loss function combining dice loss and cross entropy, and then by fine-tuning with focal loss. The model was selected from pre-studies on the challenge data.

The main contributions of this paper are two-fold. On the one hand we not only discuss our challenge model and the results, but extend our pre-studies towards a comprehensive comparative overview of additional architectures and loss functions for the segmentation of Arabidopsis root images. On the other hand, we present our approach for post-processing the segmentation results towards proper reconstruction of the complete root system architecture (RSA), focusing on the challenge task of extracting the MRs. Root segmentation results and outcomes of MR extraction are comprehensively evaluated quantitatively as well as qualitatively on the challenge training and test data sets.

The remainder of this paper is organized as follows. In Sec. 2 we give an overview of related work before we present our studies and methods in Sec. 3. Details about the data sets are provided in Sec. 4, while experimental results are presented in Sec. 5. A conclusion is given in Sec. 6.

2. Related Work

Deep learning and CNNs became the prevailing technique in image analysis with the publication of AlexNet [11] for the task of object classification. In the following years they have been extended to the task of semantic segmentation with the proposal of FCN in [13] and encoder-decoder architectures with SegNet [2] and U-Net [18]. These architectures have been extended based on residual-blocks [7], inception modules [24], and the hour glass architecture [16], see also [5] for an early review.

Besides network architecture another important ingredient for good performance is training with an adequate loss function. [8] discuss and evaluate several common ones like cross entropy and dice loss, and also less common ones like focal loss. These functions have the drawback to penalize false positives or false negatives without considering distances to the nearest annotated and the next predicted foreground pixel, respectively. The Weighted Hausdorff Distance (WHD) is proposed in [27] to overcome this problem.

The task of segmenting roots from images is often tackled with conventional segmentation techniques like intensity thresholding in BRAT [22] or ridge filtering in MyRoot [6], sometimes still relying on manual user intervention, e.g., for selecting appropriate threshold values like in EZ-Root-VIS [19]. Recently CNNs gained larger importance in this field. SegRoot [25] adopts the SegNet architecture and applies a dice loss function for extracting roots from minirhizotron images. [23] build upon the U-Net architecture with a loss combining cross entropy and the dice loss for the same task. RootNav 2.0 [28] is specifically designed for assay images using an encoder-decoder configuration integrating an hourglass network at the interface between encoder and decoder. PhenomNet [29] not only tackles the root segmentation task, but also integrates Recurrent Neural Networks based on Long Short-Term Memory to couple phenotypic predictions with genotypic analysis. In [4] a CNN based on U-Net employing residual blocks is proposed. It applies deep supervision of intermediate results and adds convolutional layers at the end of the U-Net core.

The separation of roots and background is only the first step in root image analysis. In many cases subsequent post-processing steps are applied, e.g., to close gaps and link segmented components which belong to the same root. In MyRoot 2.0 [6] a tracking algorithm is implemented which aims to link all fragments of a root between root tips and the hypocotyl based on distance heuristics. RootNav [17] adopts the A^* search algorithm to extract paths from root tips to seeds along lateral and MRs. While in the original paper [17] seeds and tips had to be selected manually by the user, in RootNav 2.0 [28] these are now automatically predicted by the CNN in parallel to potential root pixels.

3. Methods

3.1. Semantic Segmentation

For our studies we choose the basic encoder-decoder variants SegNet and U-Net due to their popularity especially in the life sciences and their use for the root segmentation task [23, 25]. They differ mainly in how they incorporate information from the encoder stage into the decoder. In SegNet, the positions of maximal values selected in the max-pooling operations are used in the corresponding upsampling step in the decoder to initialize the upsampled feature maps. The other values are filled with zeros and then interpolated with convolution. In contrast, U-Net upsamples

the last feature map in a resolution level with a trainable transpose convolution and concatenates the last feature map from the encoder with the same spatial resolution.

In addition we investigate hierarchical feature integration (Hi-Fi) proposed in [31] for the task of skeleton detection in the wild. This can be viewed as an extension of FCN [13] and Holistically-Nested Edge Detection (HED) [26]. The latter uses the last feature map of all resolution levels to compute multiple predictions, the so-called side outputs, which are subject to intermediate supervision. In addition they are fused to the final prediction. Hi-Fi proposes a richer way to incorporate the features of the encoder. First, not only the last, but all feature maps of each resolution level are fed into the side outputs. Second, the feature maps are not directly fused into prediction with a 1×1 kernel, but first convolved with 3×3 kernels into features maps as a basis for side output predictions. Third, features from neighboring resolution levels are combined, which results in Hi-Fi level 1. This combination of neighboring resolutions may be recursively repeated yielding further Hi-Fi levels and, as in HED, all side outputs are supervised. [31] advise to use one or two levels of the hierarchy. The tasks of edge or skeleton detection and semantic segmentation share common challenges. E.g., HED, proposed for edge detection, was applied to skeleton detection [10, 20, 31]. In previous studies we found HED suitable for root detection in minirhizotron images. As in addition roots exhibit strong symmetries we explore the potential of Hi-Fi for root segmentation.

A second focus in our study are loss functions as they optimize different characteristics of the segmentation. These are the cross entropy (CE) commonly used for semantic segmentation and the dice loss (DI) [14] as the inverse of the dice score which optimizes one of the evaluation metrics. In addition we combine DI with CE weighted by 0.3 (CombCED) as suggested in [23] to overcome a drawback of DI yielding a zero loss if no pixel is annotated as root.

We also use the Weighted Hausdorff Distance (WHD) defined in [27] as:

$$\begin{aligned} L_{\text{WHD}} = & \quad \frac{1}{\left|\tilde{Y}_+\right| + \varepsilon} \sum_{b\in\Omega} p_f(b) \min_{a\in Y_+} d(b,a) \\ & + \quad \frac{1}{|Y_+|} \sum_{a\in Y_+} \min_{b\in\Omega} \frac{d(a,b)+\varepsilon}{(p_f(b))^\alpha + \frac{\varepsilon}{d_{\max}}}, \end{aligned} \tag{1}$$

where Y_+ is the set of annotated foreground pixels, $p_f(b)$ the probability of pixel b to be predicted as foreground, Ω the image domain, $|\tilde{Y}_+| = \sum_{b\in\Omega} p_f(b)$, $d(a,b)$ the Euclidean distance between two pixels, α a weighting factor, and $d_{\max}$ the maximal distance between two pixels. The first term is a proxy to the average distance of predicted foreground pixels to the nearest annotated foreground pixel. $\varepsilon = 10^{-6}$ is added to avoid division by zero. The second term is an approximation of the averaged minimal distance of foreground pixels to the nearest prediction. Setting $\alpha>1$ emphasizes the second term with respect to the first one.

The WHD as defined is vulnerable in case no or few pixels are annotated as foreground and the majority of pixels is predicted as background with a large probability. Although this prediction is near the correct answer the first term yields a large value. We cure this problem setting predictions $p_f(b) < 0.1$ to zero. If no pixel is annotated as foreground $d(a,b)$ in the first term is undefined and we define $d(a,b)$ as the minimal distance of b to the border of Ω plus one. The second term is defined as zero in this case.

In [27] it is reported that WHD leads to unstable training and is therefore combined with the patch-based point loss (PPL) defined as

$$L_{\text{PPL}} = \sum_{\Omega_{i,j}} \left| \sum_{b\in\Omega_{i,j}} \tilde{p}_f(b) - |\Omega_{i,j} \cap Y_+| \right| \tag{2}$$

where the windows $\Omega_{i,j}$ are a partitioning of the image domain, and $\tilde{p}_f(b)$ is the predicted probability $p_f(b)$ if $p_f(b) > \lambda_T$, zero otherwise. Thus, PPL compares the sum of these clamped probabilities and the number of foreground pixels in a window summed over all non overlapping windows. Similar to [27] we use a linear combination

$$\mu \cdot WHD + (1-\mu)\cdot PPL, \qquad \mu \in [0,1], \tag{3}$$

as loss function after an initial training, where in our experiments we use CombCED.

In analogy we use the focal loss (FL) [12] as a loss function subsequent to an initial training of weights in an attempt to improve a pretrained network. FL generalizes CE by adding a modulation factor:

$$L_{\text{FL}} = -\sum_{b\in\Omega} (1-p_t(b))^\gamma \log(p_t(b)), \tag{4}$$

where $p_t(b)$ is the predicted probability for the true class. In case of $\gamma = 0$, FL reduces to CE. FL emphasizes the hard to predict examples during training.

3.2. Post-processing and main root extraction

Applying the CNNs to the input images provides us with binary predictions for root pixels and background. However, as until now no topological knowledge has been considered in the segmentation process, roots sometimes decompose into multiple connected components. While this is not a serious problem with regard to overall segmentation accuracy as, e.g., measured by recall and precision rates, gaps significantly hamper the extraction of overall root system architecture and, with regard to the challenge task, in particular the extraction of the MR for each plant.

Therefore we use a post-processing pipeline on the segmentation results where we also consider temporal information from time series. The post-processing consists of two

main stages. First we aim at closing gaps in roots and reconnect branches to the main component of a plant that might have been detached. Subsequently, assuming that each plant is now represented by a single connected component, the MR of the corresponding plant is extracted based on topological skeleton analysis and graph-based path search. Both stages rely on the skeletons of the connected components which are extracted with the algorithm of Zhang et al. [30].

Gap closing and branch reconnection Gaps splitting the segmentation of a root system can be distinguished into two main categories: (I) gaps within a stretch of a root, (II) gaps disconnecting the root system at branching points. We consider both cases in turn.

Gaps of type (I) are characterized by pairs of end points in the skeleton where both points are only a short distance apart from each other and the skeleton segments located next to the end points do not significantly differ in their orientations. Thus, we initially locate skeleton end points as points with not more than one neighboring pixel in the skeleton. Subsequently all pairs of end points are detected which satisfy the following three criteria: (i) small distance, (ii) similar orientation of the skeletons at both end points, (iii) which are in turn similar to the orientation of the line connecting both end points. Thresholds for these criteria are set empirically. To connect such pairs of end points we apply a Dijkstra shortest path search. To this end we convert the local image patch around both end points into a graph representation with the pixels as graph nodes and their 8-neighbors connected by edges. The weight of the edges is defined as the response of an anisotropic vesselness filter measuring the correlation between the local intensity structure and the theoretical landscape of locally linear root structures (for more details see [15] where the same idea is used to close cell contours). If the final path is not longer than 1.5 times the distance between the end points they are connected. The width of the connecting segment is derived from the width of the root segments to be connected.

In case of type (II) gaps typically only a skeleton end point exists in the detached branch, but not in the root to be connected to. To check if an end point of a branch should be reconnected to a nearby root component we estimate the orientation of the final part of the skeleton branch and search in its direction for nearby root pixels. If at least one pixel is found within a maximum distance we insert a line segment and derive its width from the width of the branch.

Main root extraction The root system of an Arabidopsis plant consists of a single MR and any number of lateral roots. According to biological experts there is no clear definition of the MR except that it starts at the hypocotyl and is usually the longest root. In many images of the challenge data sets the hypocotyl cannot easily be localized, as it is often hidden by leaves and their stems. Thus, we define it as the end or branch point of the skeleton of the plant component located topmost in the image. As tip of the MR we use the pixel of the component closest to the bottom image border. The MR can then be found as the shortest path between root tip and hypocotyl applying a Dijkstra path search.

Obviously this approach assumes that each plant is represented by a single component. In practice this assumption is often not fulfilled as even after gap closing multiple components may survive for one root system, and further components may result from clutter or leaves. Hence, to select a single component per plant we define position priors for the hypocotyls and restrict locations of tip points with positions in the previous frame. In all experiments of the challenge test data set four seeds are initially planted in the upper third of the images at approximately constant positions. Thus, we define four regions of interest (ROIs) in the upper third of the image around the four seed positions and process each of these ROIs independently. We identify the largest connected component within the ROI and detect the MR within this component as described above.

The localization of the tip of the MR as described fails in some cases. One such situation occurs if the roots of two or more plants overlap and the corresponding components merge. We detect such cases by comparing the size of the components of each plant between subsequent frames. In case of overlapping roots the size usually almost doubles and the connected component covers more than one plant. As a consequence it contains the tip points of several MRs and the lowest pixel selected as tip may be the wrong one. We avoid this by enforcing additional constraints on the tip point of a plant, i.e., enforce a certain maximal distance from the position in the previous frame. This maximum distance, however, may lead to cases where no tip is found at all. This happens if a wrong connected component was initially selected, e.g., due to clutter in the ROI. In such cases we process the second largest component which in the majority of situations is the correct one.

Finally we remove all components which are too small and/or too far away from any MR component (see Sec. 5.2).

4. Data Sets

We conduct our experiments on the data sets of the Arabidopsis Root Segmentation Challenge 2021. For model training two data sets with partial annotations were provided. Due to time constraints we use only the one with binary labels consisting of 34 video sequences yielding a total of 1542 images. For 207 images annotations of roots and background is given, yielding data set D_{bin}. The challenge test data set D_{test} provided without annotations comprises 22 video sequences with a total number of 933 images.

We randomly partition D_{bin} into training D_{bin}^{train}, validation D_{bin}^{val} and test D_{bin}^{test} data with the ratio of 70:20:10. We crop images to size 512×512 pixels with an offset of 384 or to size 256×256 pixels with an offset of 192.

5. Results

5.1. Semantic Segmentation

Experimental Setup For SegRoot training we use the SegNet implementation[1] and optimizer described in [25].

The U-Net architecture in the original structure and in the VGG16 structure as well as the Hi-Fi architecture are our own implementations in PyTorch. Training and test is based on the code[2] used in [23]. We implement FL as well as WHD and PPL in addition to the existing dice and CE losses. In PPL we set the patch size to 32, as suggested in [27], and $\lambda_T = 0.5$ as this is our threshold for prediction. During training we use the stochastic gradient descent as optimizer with a momentum 0.99 and weight decay 10^{-5}. As a result we always choose the epoch with the best validation results and also test our net on this epoch.

SegRoot The SegRoot network [25] was initially designed for root segmentation in minirhizotron images. Here we retrain the network from scratch in different configurations. The SegRoot code supports to configure the network structure, i.e., the number of feature maps in the first convolutional layer (width), which is then doubled subsequent to each max pooling, and also the number of resolution levels (depth). In the original work a width of 8 and a depth of 5, i.e., an 8-5-net, was proposed as best compromise between network complexity and segmentation performance. Configurations with larger widths showed slightly better performance, but generally required more training time.

We tested configurations with widths of $8, 16$ and 32, and depths of 4 and 5. A learning rate $lr = 10^{-2}$ was applied for the 8-5-net in [25], and for wider configurations $lr = 10^{-4}$ was chosen. For widths of 16 and 32 $lr = 10^{-4}$ worked well in our studies, but with $lr = 10^{-2}$ no learning process was observed on the 8-4- and 8-5-nets, i.e., scores dropped instantly close to zero and never recovered. Thus, we tested also learning rates of $10^{-3}, 10^{-4}$ and 10^{-5}, but with $lr = 10^{-3}$ learning neither happened. Batch sizes varied with available memory from 64 for the 8-5- and 16-4-nets, 32 for the 16-5- and 32-4-nets to 16 for the 32-5-net. SegRoot requires to use crops of size 256×256. The dice scores for the various experiments on D_{bin}^{test} are shown in Tab. 1.

The best dice score on D_{bin}^{test} was achieved with the 32-4-net performing slightly better than 16-4 with $lr = 10^{-4}$ and 32-5 with $lr = 10^{-5}$. Configurations like 8-4 and 16-5 with $lr = 10^{-4}$ performed also well. On the contrary, trainings with $lr = 10^{-5}$ performed significantly worse, except for the 32-5-net, but with $lr = 10^{-4}$ no training at all could be initiated for this net. According to studies on configurations with a width of 64 in [25] it might be hypothesized that such network models could boost the SegRoot performance further, and we plan to extend our study towards these models.

[1] https://github.com/wtwtwt0330/SegRoot

[2] https://github.com/Abe404/segmentation_of_roots_in_soil_with_unet

	depth = 4		depth = 5	
width	$lr = 10^{-4}$	$lr = 10^{-5}$	$lr = 10^{-4}$	$lr = 10^{-5}$
8	0.833	0.614	0.825	0.699
16	0.870	0.776	0.847	0.781
32	0.874	0.817	fail	0.870

Table 1. Dice scores for D_{bin}^{test} for different SegRoot widths (8, 16, 32), depths (4, 5) and learning rates ($lr = 10^{-4}$,10^{-5}). For the 32-5-net with $lr = 10^{-4}$ no learning process could be observed.

U-Net and Hi-Fi As an alternative to SegNet we investigate U-Net and Hi-Fi. As the structure of the encoder resp. backbone we employ VGG16 [21] and the one proposed in the original U-Net paper [18]. As loss functions CE, DI, and CombCED are used, group normalization as opposed to no normalization, and learning rates of 10^{-3} and 10^{-4}. Thus, a total of 48 combinations of hyper parameters result.

First, we conducted 8 replicates of the hyper parameter combination used for the challenge submission (see below) to estimate the variance due to random initialization. The mean dice score on D_{bin}^{test} is 0.910 with a maximal difference of 0.0020. In the following we consider differences of the dice score in the second decimal place as considerable and not to be attributed to random effects.

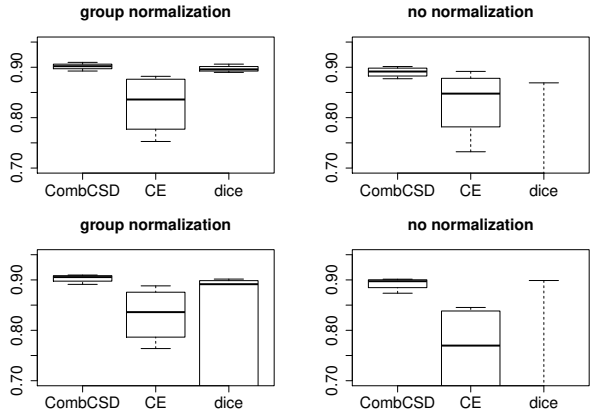

Figure 2. Boxplot of dice scores on D_{bin}^{test} for different backbone structures, loss functions, and normalizations. Top row: VGG16 structure, bottom row: original U-Net structure.

The boxplots in Fig. 2 show that CombCED yields a better dice score in almost all cases compared to CE. This may be partly attributed to the fact, that we assess performance with the dice score which is part of the CombCED loss function. In addition, however, CE is less robust with respect to hyper parameters which are thus more difficult to tune. DI can be expected to perform well, as it coincides with the performance measure. However, training is quite unstable in this case resulting in a dice score of less than 0.02 for 8 of the experiments. As evident from Tab. 2 only one of

lr	CombCED	CE	DI
U-Net, VGG16 Structure			
10^{-3}	0.902	0.882	0.895
10^{-4}	0.903	0.802	0.900
U-Net, Original U-Net Structure			
10^{-3}	0.910	0.890	0.890
10^{-4}	0.904	0.809	0.900
Hi-Fi, VGG16 Structure			
10^{-3}	0.910	0.870	0.906
10^{-4}	0.893	0.753	0.890
Hi-Fi, Original U-Net Structure			
10^{-3}	0.907	0.863	0.902
10^{-4}	0.891	0.764	0.006

Table 2. Dice scores on D_{bin}^{test} using group normalization.

these instable trainings occurs when using group normalization. In contrast we find only one such experiment for CE, and none for CombCED. In case of successful training in most cases competitive dice scores result compared to CombCED, which is true for all experiments with group normalization and VGG16. We speculate that CE alleviates the instability of DI in the combination still giving the advantage of dice as the performance measure.

Next we analyze the effect of group normalization. Fig. 2 indicates that adding normalization tends to produce a more stable performance. Comparing all experiments with respect to normalization variants, CombCED gives a considerable better dice score in two thirds of experiments and comparable results otherwise. For CE no clear tendency can be observed if the single experiment with a lack in proper learning is omitted. Due to these findings we only consider training with group normalization in the following.

Tab. 2 shows that using CombCED the four combinations of network type and backbone structure perform comparable with the exception of Hi-Fi and $lr = 10^{-4}$. Training with DI yields comparable results except for one experiment where no learning happened at all. This dice score of 0.90 ± 0.01 for these experiments is the best performance on our test set we observed. The CE delivers considerable worse dice scores for most of the cases, especially for $lr = 10^{-4}$. With respect to learning rate $lr = 10^{-3}$ outperforms 10^{-4} for several combinations. However, we feel that more experiments should be performed and expect that the appropriate one depends on the other hyper parameters.

In all experiments where learning was successful precision and recall vary slightly and quite symmetric around the stable dice score. In Fig. 3 the evolution of performance during training on D_{bin}^{val} is displayed.

In summary, we find a slight advantage of CombCED loss and group normalization, while both network types – U-Net and Hi-Fi – and backbone structures perform comparable. Obviously, still more combinations of hyper parameters could be examined. However, we speculate that no significant improvements may be achieved especially taking ambiguities of groundtruth annotation into account, see also "Qualitative results" below.

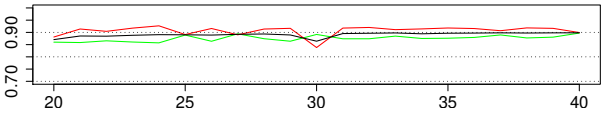

Figure 3. Performance on D_{bin}^{val} in the course of training. Black: dice score, red: precision, green: recall.

μ	$lr = 10^{-8}$	$lr = 10^{-9}$	$lr = 10^{-10}$
0.4	0.846/0.883/0.864	0.859/0.886/0.872	0.862/0.896/0.879
0.5	0.844/0.845/0.845	0.864/0.888/0.876	0.862/0.895/0.878
0.6	0.818/0.601/0.693	0.863/0.891/0.877	0.862/0.896/0.879

Table 3. Test results on training with a linear combination of WHD and PPL with varying weighting factor μ as loss function and different learning rates. For $lr = 10^{-8}$, 10^{-9} and 10^{-10} recall/precision/dice are given.

lr	Recall	Precision	Dice
10^{-7}	0.856	0.946	0.899
10^{-8}	0.889	0.929	0.908
10^{-9}	0.907	0.914	0.910
10^{-10}	0.908	0.912	0.910

Table 4. Test scores for focal loss with $\gamma = 1.0$ trained on pretrained weights with different learning rates.

WHD and PPL loss We employ a linear combination of WHD and PPL to train a U-Net with the VGG16 structure which was pretrained using CombCED, $lr = 10^{-4}$ and group normalization. Due to memory limitations caused by WHD crops sized 256×256 pixels are used and α is set to 4. We train 15 epochs and then choose the epoch with the best validation dice score for testing and present results in Tab. 3. Training with $lr = 10^{-9}$ and $lr = 10^{-10}$ results in a slight decrease of the dice score compared to the pretrained network and a slight imbalance between precision and recall developments. Using $lr = 10^{-8}$ intensifies this effect especially with increasing μ from 0.4 to 0.5 and 0.6. To summarize, at least based on these three performance metrics WHD does not give an improvement, but rather a decline in performance. Potential improvements with respect to the aim of geometry-awareness are hard to quantify and need to be scrutinized more carefully.

Focal loss As a second loss to further train the same U-Net with VGG16 structure we use the focal loss. In Tab. 4, we show recalls, precisions and dice scores for different learning rates. With respect to these performance measures $lr = 10^{-10}$ and 10^{-9} are obviously too small

to make a difference. With the increase of the learning rate, precision and recall diverge with considerable higher precision while reducing the recall. This leads to an insignificant change of the dice score. With $lr = 10^{-7}$ the imbalance of recall and precision further increases and the decrease of the dice score gets considerable. Whether this increase in precision for the price of smaller recall is an advantage and if so to which degree is to be answered by the application.

Qualitative results For our challenge results we choose the U-Net with VGG16 structure, CombCED, $lr = 10^{-4}$, and group normalization due to best performance in our preliminary tests on D_{bin}^{test} (data not shown). After 40 epochs we continued training with FL setting $\gamma = 1.0$ using $lr = 10^{-8}$ as visual inspection indicated superior performance on D_{test}. We observed that including training with FL decreases false positives (FPs) in the leaf regions. While it also induces more false negatives (FNs) in root gaps this is at least partially compensated by gap closing in the post-processing stage. Note, that we trained using crops of size 256×256 during the preliminary tests, thus the performances given in Tab. 2 and 5 slightly differ.

Qualitative aspects of the results achieved with this network are discussed next. While the roots are usually quite well segmented in the middle and lower parts of the images, errors seem to appear more frequently in the upper parts containing the leaves. To validate this quantitatively, we evaluate the region at the top including most leaves and the rest of the images separately. The leaf region is sized 2000×645 pixels and located with its top left corner at position $(620, 0)$. In Tab. 5 recalls, precisions, and dice scores are given.

	Recall	Precision	Dice
Image complete	0.884	0.940	0.911
Leaf region	0.844	0.912	0.879
Non-leaf region	0.902	0.951	0.926

Table 5. Evaluation results on leaf and non-leaf regions in D_{bin}^{test}.

Performance is best in non-leaf regions and outperforms all scores on the complete images. For the leaf regions performance significantly drops compared to the non-leaf regions, but also with regard to complete images, e.g., the dice score drops from 0.926 and 0.911, respectively, to 0.879. Thus, improving segmentation in particular in leaf regions seems to be promising for boosting segmentation quality.

Given the above observations we further investigated root segmentation in leaf regions by visual groundtruth comparisons. It turns out that root annotations seem inconsistent sometimes which occasionally causes errors with regard to groundtruth although the segmentation appears reasonable according to the image data. First, roots covered by leaves are sometimes annotated as foreground, sometimes not. For the samples in Fig. 4 our network predicts parts of the roots behind the leaves, which for the example in the left column (A) results in FNs as well as TPs, while for the example in the second column (B) FPs result. Second, the hypocotyl position is sometimes very roughly localized so that parts of the stem are marked as root. In Fig. 4, third column (C), the hypocotyl is properly localized and our segmentation is consistent with the annotation, while in the forth column (D) parts of the stem are also annotated as foreground which we miss in our segmentation.

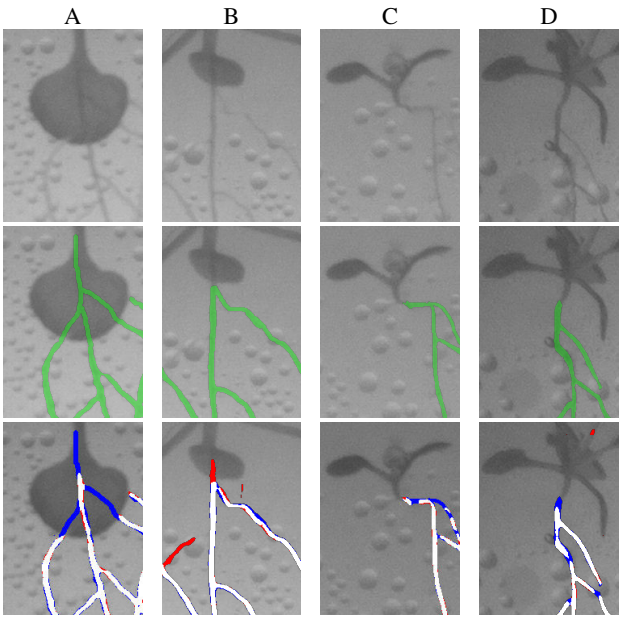

Figure 4. Examples for inconsistent annotations of roots covered by leaves (columns A and B), and examples for rough hypocotyl localization (columns C and D). Top row: input images, middle row: groundtruth, bottom row: overlay of our segmentations (white: true positive, red: FP, blue: FN).

Additionally, in some images there are spurious pixels annotated as foreground distant to the nearest root system. While these incorrect annotations have only weak impact on the dice score the effect on the Hausdorff Distance, one of the challenge metrics, may be considerable.

5.2. Root Segmentation Challenge

According to the challenge organizers our CNN described in the previous paragraph achieved a dice score of 0.761 and completeness and correctness scores of 0.894 and 0.955, respectively, on a subset of 132 images of D_{test} which were used for producing the challenge results. Compared to a human annotator who provided annotations for comparison and achieved a dice score of 0.802 with completeness and correctness scores of 0.957 and 0.948, respectively, this is only slightly worse. However, if we compare to the dice scores on D_{bin}^{test} of roughly 0.91 that we observed

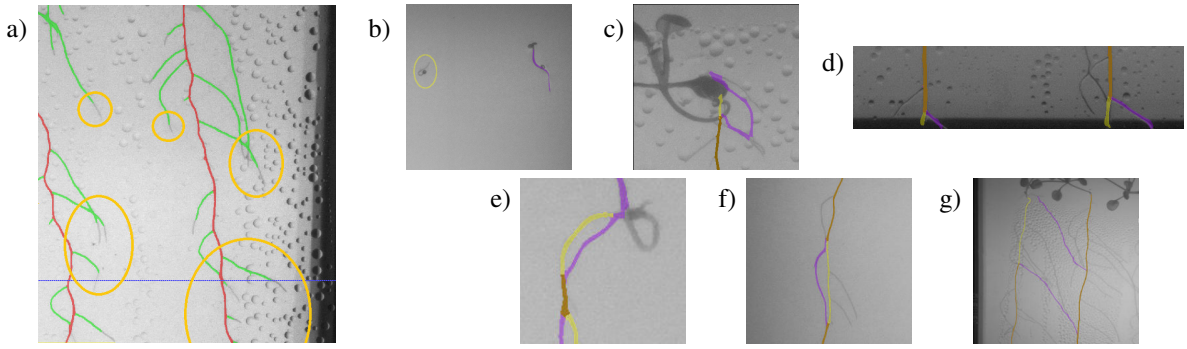

Figure 5. a), Sample segmentation result on D_{test}. Tip segments of the lateral roots are frequently missed (red: TP of MRs, green: TP of lateral root). Image courtesy to the Root Segmentation Challenge Team. b)-g), examples for errors in MR root segmentation according to our own judgement of groundtruth (orange: TP, yellow: FN, purple: FP). For further details refer to the text.

during training of our model, the challenge results are significantly worse. The primary reason for these large deviations seem to originate from the fact that we often miss larger parts of lateral root tips as shown in Fig. 5 a. This could point to significant differences in image characteristics between our training data D_{bin} and the test data D_{test}.

Our post-processing stage for extracting the MRs relies on several empirically chosen parameters which were set rather liberal. As the challenge evaluation metrics considers the total number of detected components, our intention was to reduce the number of components as much as possible while accepting some small erroneously closed gaps. Hence, we extract paths between end points with a maximum distance of 40 pixels and a maximum deviation in orientations of $50°$. For reconnecting branches the maximal distance is set to 20 pixels. After extracting the MR we remove all components smaller than 75 pixels, components smaller than 250 pixels if more distant than 100 pixels to all plant components, and components smaller than 500 pixels if more distant than 350 pixels. This allows to eliminate clutter, but keep branches and other root parts that could not be linked to the main component of a plant.

We performed a thorough analysis of the quality of our MR extraction by visually inspecting all 933 images of D_{test}. In 17 out of the 22 video sequences almost 90% of the MRs seem to be identified correctly. This is also confirmed by our challenge results where we achieve a completeness of 0.918 and a correctness of 0.952 on the MR pixels. If MR extraction fails to a large extent this is mostly attributable to one of three typical issues. In the early images of a video sequence where plants start to grow it is often hard to distinguish between small components resulting from clutter and correct root components (Fig. 5 b). From our subjective assessment and without knowing the groundtruth we may sometimes miss small roots and MRs in the first images of a sequence. Likewise, sometimes wrong components are selected as root components leading to MRs located in noise components. A third more serious source of errors are path errors where the correct component is traced, however, the MR path includes wrong root segments. This mainly happens at the top or bottom of root components if the hypocotyl or tip point is wrongly detected, e.g., due to the roots growing out of the image or plate, or the seed points and roots being covered by parts of a leaf or stem (Fig. 5 c, d). In some rare cases the MR path is wrongly extracted due to ambiguities in the image data (Fig. 5 e,f) where it is hardly possible to correctly trace the MR without considering additional temporal information. Including such data in MR extraction would be one of the most promising directions for improving the quality of MRs. Anyway, most of the MR errors due to these issues affect only small portions of the MR. In three video sequences, however, MR extraction fails seriously for a couple of images. In these sequences components for different plants merge and the path extraction in parts follows roots of the wrong plant (Fig. 5 g). Here also temporal information and path alignment between successive time points might help to extract MR paths more robustly.

6. Conclusions

CNNs are a common approach for semantic segmentation and have also gained interest to segment root images. Here we present our approach for the segmentation task of this year's CVPPA Arabidopsis Root Segmentation Challenge. In general, we achieve fair segmentation scores and particularly succeed in identifying the main roots, while complete segmentation of the tips of lateral roots remains challenging. Thus, our study on alternative CNN architectures, loss functions and parameters could guide the development of more powerful models. Together with considering additional temporal information from time series in the post-processing stage this may lead to further performance boosts and foster progress in root image segmentation.

References

[1] Jonathan A. Atkinson, Michael P. Pound, Malcolm J. Bennett, et al. Uncovering the hidden half of plants using new advances in root phenotyping. *Current Opinion in Biotechnology*, 55:1–8, 2019.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[3] Hans de Kroon, Liesje Mommer, and Aya Nishiwaki. Root competition: towards a mechanistic understanding. In *Root Ecology*, pages 215–234. Springer, 2003.

[4] Nicolás Gaggion, Federico Ariel, Vladimir Daric, et al. ChronoRoot: High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. *bioRxiv*, 2020. DOI: 10.1101/2020.10.27.350553.

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, et al. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

[6] Alejandro González, Xavier Sevillano, Isabel Betegón-Putze, et al. MyROOT 2.0: An automatic tool for high throughput and accurate primary root length measurement. *Comp. and Electronics in Agriculture*, 168:105125, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Deep residual learning for image recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Shruti Jadon. A survey of loss functions for semantic segmentation. In *Proc. of the IEEE Conf. on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–7, 2020.

[9] Mark G. Johnson, David T. Tingey, Donald L. Phillips, et al. Advancing fine root research with minirhizotrons. *Environmental and Experimental Botany*, 45(3):263–289, 2001.

[10] Wei Ke, Jie Chen, Jianbin Jiao, et al. SRN: Side-output residual network for object symmetry detection in the wild. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1068–1076, 2017.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, et al. Focal loss for dense object detection. In *Proc. of the IEEE Int. Conf. on Computer Vision*, pages 2980–2988, 2017.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[14] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. of Fourth IEEE Int. Conf. on 3D Vision*, pages 565–571, 2016.

[15] Birgit Möller and Katharina Bürstenbinder. Semi-automatic cell segmentation from noisy image data for quantification of microtubule organization on single cell level. In *Proc. of IEEE 16th Int. Symposium on Biomedical Imaging (ISBI)*, pages 199–203, Venice, Italy, April 2019.

[16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conf. on Computer Vision*, pages 483–499. Springer, 2016.

[17] Michael P. Pound, Andrew P. French, Jonathan A. Atkinson, et al. RootNav: Navigating Images of Complex Root Architectures. *Plant Physiology*, 162(4):1802–1814, 2013.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[19] Zaigham Shahzad, Fabian Kellermeier, Emily M. Armstrong, et al. EZ-Root-VIS: A Software Pipeline for the Rapid Analysis and Visual Reconstruction of Root System Architecture. *Plant Physiology*, 177(4):1368–1381, 2018.

[20] Wei Shen, Kai Zhao, Yuan Jiang, et al. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 222–230, 2016.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2015.

[22] Radka Slovak, Christian Göschl, Xiaoxue Su, et al. A Scalable Open-Source Pipeline for Large-Scale Root Phenotyping of Arabidopsis. *The Plant Cell*, 26(6):2390–2403, 2014.

[23] Abraham G. Smith, Jens Petersen, Raghavendra Selvan, et al. Segmentation of roots in soil with U-Net. *Plant Methods*, 16(1):1–15, 2020.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[25] Tao Wang, Mina Rostamza, Zhihang Song, et al. SegRoot: A high throughput segmentation method for root image analysis. *Comp. and Electronics in Agriculture*, 162:845–854, 2019.

[26] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. of the IEEE Int. Conf. on Computer Vision*, pages 1395–1403, 2015.

[27] Weijian Xu, Gaurav Parmar, and Zhuowen Tu. Geometry-Aware End-to-End Skeleton Detection. In *Proc. of British Machine Vision Conf.*, volume 2, page 7, 2019.

[28] Robail Yasrab, Jonathan A. Atkinson, Darren M. Wells, et al. RootNav 2.0: Deep learning for automatic navigation of complex plant root architectures. *GigaScience*, 8(11), 2019.

[29] Robail Yasrab, Michael P. Pound, Andrew P. French, et al. PhenomNet: Bridging Phenotype-Genotype Gap: A CNN-LSTM Based Automatic Plant Root Anatomization System. *bioRxiv*, 2020. DOI: 10.1101/2020.05.03.075184.

[30] Tongjie Y. Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.

[31] Kai Zhao, Wei Shen, Shanghua Gao, et al. Hi-Fi: Hierarchical Feature Integration for Skeleton Detection. In *Proc. of. Int. Joint Conf. on Artificial Intelligence*, pages 1191–1197, 7 2018.