Pattern Recognition Letters 183 (2024) 125–132

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Patch-based probabilistic identification of plant roots using convolutional neural networks

A. Cardellicchio [a], F. Solimani [a], G. Dimauro [b], S. Summerer [c], V. Renò [a,*]

[a] Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Via Amendola 122 D/O, Bari, 70126, Italy

[b] University of Bari, Department of Computer Science, Via E. Orabona, 4, Bari, 70125, Italy

[c] ALSIA Centro Ricerche Metapontum Agrobios, s.s. Jonica 106, km 448.2, Metaponto, 75010, Italy

ARTICLE INFO

Editor: Maria De Marsico

Dataset link: https://github.com/vitoreno/pyroots.git

Keywords:
Deep learning
Root system architecture
Convolutional neural network
Computer vision

ABSTRACT

Recently, computer vision and artificial intelligence are being used as enabling technologies for plant phenotyping studies, since they allow the analysis of large amounts of data gathered by the sensors. Plant phenotyping studies can be devoted to the evaluation of complex plant traits either on the aerial part of the plant as well as on the underground part, to extract meaningful information about the growth, development, tolerance, or resistance of the plant itself. All plant traits should be evaluated automatically and quantitatively measured in a non-destructive way. This paper describes a novel approach for identifying plant roots from images of the root system architecture using a convolutional neural network (CNN) that operates on small image patches calculating the probability that the center point of the patch is a root pixel. The underlying idea is that the CNN model should embed as much information as possible about the variability of the patches that can show chaotic and heterogeneous backgrounds. Results on a real dataset demonstrate the feasibility of the proposed approach, as it overcomes the current state of the art.

1. Introduction

Over the recent years, computer vision has become one of the main assets for plant phenotyping studies, starting from the analysis of remote sensing data of crops and fields to the development of semi-automated software that could help domain experts in the analysis of Root Systems Architectures (RSAs) [1]. Plant phenotyping is intended to perform non-destructive analysis of complex plant traits related to growth, yield, and adaptation to stress with an elevated degree of accuracy and precision. Often, such tasks are performed by human operators, subject to limitations in experience and skills. Furthermore, recent years have seen the deployment of several High-Throughput Platforms (HTPs), which are able to provide a relevant stream of data concerning the phenotypical traits of plants. Consequently, human operators also experienced an increased workload, which resulted in the requirement for automatic and non-biased protocols to overcome these issues.

The literature shows some examples of machine learning (ML) and deep learning (DL) applications in specific fields of RSA segmentation. Still, the common bottleneck reported in the state-of-the-art works is the lack of generalized and standardized data that could be used to train extremely complex deep learning architectures. In other words, as these models generally require a high number of labeled samples to be trained, a strong limitation in their practical applicability is represented by data availability.

Unfortunately, data sampling and labeling is a time-demanding process, biased by the skills and workload of the human operator. Hence, obtaining meaningful data that artificial intelligence models can use can be difficult. Consequently, one of the first issues related to the effective use of artificial intelligence models in the plant phenotyping field concerns data availability, as common datasets used in object recognition, such as ImageNet, are not specifically designed to work with domain-specific data.

To solve this issue, this work introduces a processing pipeline for the end-to-end analysis of RSAs of plants. Specifically, image segmentation starts with patches extracted automatically from labeled images acquired using the method described in our previous work [2]. The pixel-based extraction criterion generates a large number of patches from a relatively limited amount of images of RSAs, hence lowering the burden on human operators for data gathering and labeling.

* Corresponding author.
E-mail addresses: angelo.cardellicchio@stiima.cnr.it (A. Cardellicchio), firozeh.solimani@stiima.cnr.it (F. Solimani), giovanni.dimauro@uniba.it (G. Dimauro), stephan.summerer@alsia.it (S. Summerer), vito.reno@stiima.cnr.it (V. Renò).

https://doi.org/10.1016/j.patrec.2024.05.010
Received 23 December 2022; Received in revised form 11 April 2024; Accepted 15 May 2024
Available online 18 May 2024

The images obtained are then used to train a straightforward and effective convolutional neural network (CNN) to estimate the probability of observing a root pixel. The choice of the architecture behind the proposed CNN model follows a specific criterion: finding the simplest yet effective model that can be used to identify the RSAs of the plants starting from the patch extracted with the proposed system. Consequently, a relatively shallow and simple architecture was selected, with three convolutional layers, each with a limited number of filters. To further reduce the complexity of the network, a max pooling layer was added before the latest fully connected layer, which led to a binary classification layer where a sigmoid function performs the final decision using a statistically determined threshold. The results show that the model can outperform previous state-of-the-art approaches on the proposed dataset.

The rest of the paper is organized as follows. In Section 2, the current state-of-the-art is depicted. Then, Section 3 describes the methodology developed. Section 4 shows the experiments and the results achieved, while Section 5 concludes the paper.

2. Related works

RSAs are difficult to observe directly, mainly due to the soil which naturally covers them [3]. As a consequence, specific non-destructive phenotyping methods have been developed, such as the use of transparent agar or germination papers, which have proven to be efficient, especially at early growth stages, with the only disadvantage of requiring the root system to grow in artificial soil [4]. Another viable approach is the use of X-ray computed tomography [5], which allows the visualization of the root system in natural soil; however, this type of system is expensive and difficult to deploy directly on the field.

Once images of the RSA have been gathered, they should be segmented to detect the roots. However, the segmentation step is usually challenging due to the complex nature of the RSA and the low contrast between soil particles and roots. To cope with these issues, several tools have been proposed.

As an example, the authors in [6] proposed a framework named *GLO-Roots*, which exploits different types of feature-based image analysis techniques, such as local pattern recognition, global, shape, and directionality analysis, to identify and extract the characteristics of the root system, also considering gene reporters and soil moisture. Another semi-automated tool, called *GT-Roots*, is proposed in [7]; GT-Roots also applies a processing pipeline to each image that starts by extracting a Region of Interest (RoI), then converts the original image into grayscale, performs adaptive thresholding, and finally applies a morphological operator to enhance the results. GT-Roots also allows for a semi or fully-automated pipeline, where the operator can manually intervene in each intermediate processing step. Authors in [1] propose *GIA-ROOTS*, whose pipeline first performs image pre-processing via rotation, crop, and scaling. Afterward, the user is asked to select a series of relevant root system traits from a set of 19 possible choices, which are then used on the segmented image to extract the root system. Another tool is *saRIA* [8], which provides a semi-automated environment for RSA segmentation and calculating phenotypic features of the RSA. The analysis pipeline includes several pre-processing steps, such as cropping, despeckling, smoothing, and inversion of image intensity. Then, adaptive image thresholding is used to segment the image in the foreground (roots) and background (soil) using Gaussian weighted mean. Morphological filters are then used to remove noise and improve the quality of the found roots, root skeletons are computed, and RSA features are extracted from a list of 44 root traits using a pixel-wise computation.

While these tools can perform extremely well when only a few images are available, they may be inadequate when a high amount of data must be processed due to the required human intervention. Furthermore, fixed processing pipelines usually lack generalization capabilities. Tools based on deep learning have been proposed to deal with these issues. One, and probably the most well-known, of such tools is SegRoot [9], that provides a binary mask of root (white pixels) and no-root (black pixels) starting from an RSA image. SegRoot is based on a modification of SegNet [10] and uses a series of standard CNN blocks (that is, 3×3 convolutional filters followed by batch normalization and ReLU activations) in the encoder. The decoder is composed of a series of unpooling layers that perform non-linear upsampling to make the output feature maps identical to the input feature maps of the corresponding encoding layer. The main difference between SegRoot and SegNet lies in the loss function, which is a modification of the Dice coefficient [11]. Another tool based on a U-architecture is DeepLabv3+ [12], which uses an U-shaped encoder based on Xception as its backbone. The approach proposed in [13] predicts two parameters representing the vertical and horizontal centroid of root distribution to reveal the phenotypic diversity of root distribution.

Despite their effectiveness, U-shaped models may be over-complex for the classification task and provide only a binary output instead of evaluating the probability of observing root or no-root portions of an image. A traditional, stacked convolutional neural network can be used accordingly if the root segmentation problem is framed as described in Section 3.

3. Materials and methods

3.1. Dataset gathering and annotation

To gather this dataset, the procedure described in our previous work has been followed [2]. Specifically, data have been collected over 40 days from cylindrical rhizotron tubes containing 5 plants of a specific genotype. To this end, approximately 3800 snapshots have been gathered, each consisting of $k = 6$ photos of a rhizotron tube for approximately 22800 raw RGB images. Such images have been captured using a Scout sca1600-14gc camera (Basler AG) with a spatial resolution of 1234×1624 (width $\times$ height). To reduce the environmental factors that could affect image quality in uncontrolled environments (e.g. outdoor with lighting changes or using different sensors to capture the same type of data), data collection took place using the High Throughput Plant Phenomics Platform (HTP) based on a LemnaTec Scanalyzer3D system located at the ALSIA Metapontum Agrobios Research Centre. The system is equipped with an automated belt conveyor system that automatically drives the samples to an acquisition chamber so that the position, orientation and lighting conditions of each captured frame can be controlled and standardized. Raw images have been then processed, first identifying the rhizotron border, which has been then rotated to evaluate the cylinder radius via geometrical formulae. Then, images have been stitched together, completing the panorama extraction step. Afterward, images have been cleaned from the contribution provided by noise generated by the influence of light reflected on the cylindrical rhizotrones via SVD. From the original dataset, 300 images have been selected for manual data annotation. These images have been hand-traced by domain experts using the Computer Vision Annotation Tool [14] and exported into the Dataset Management Framework (Datumaro) format [15]. In order to ensure accuracy and consistency in the annotated ground truth data, the ground truth masks have been labeled by three different independent domain experts. A reliability check was then performed using a major voting procedure. In more details, a pixel of the ground truth mask was labeled as root (or no-root) if at least two independent annotators labeled it as root (or no-root). This way, we aimed at reducing subjective bias that could be introduced by a single annotator, even from an expert one. An example of the ground truth image and its corresponding ground truth is provided in Fig. 1.

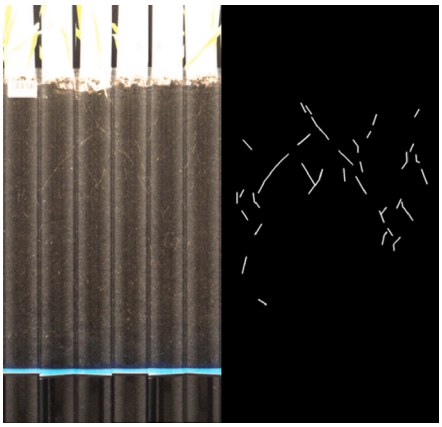


Fig. 1. On the left, a root composite image with its corresponding ground truth on the right.

3.2. Data preprocessing

Let us note that raw annotated images are not directly used for the experiments. Instead, image patches have been automatically created using an *extraction filter* of size $F_w \times F_h$. Specifically, the filter "flows" across each pixel in the image, starting from the top left towards the bottom right, provided that:

$$x_P \in \left[I_w + \frac{F_w}{2}, I_w - \frac{F_w}{2} \right] \wedge y_P \in \left[I_h + \frac{F_h}{2}, I_h - \frac{F_h}{2} \right] \quad (1)$$

In Eq. (1), (x_P, y_P) represents the coordinates of the pixel P, while I_w and I_h represent the width and the height of the original image, respectively. In other words, the filter extracts a series of patches provided that its borders completely fit within the original image, hence not applying any padding operation.

Once each patch has been extracted, its corresponding ground truth has been used to automatically label it as either a *positive* sample (that is, its center represents a root pixel) or a *negative* sample (that is, its center does not represent a root pixel).

Let us note that a dataset extracted this way could be highly imbalanced, presenting a larger number of negative samples. For this reason, during the image patches automatic extraction, a certain number of root patches from an image have been extracted. Then the same number of no-root patches is randomly selected by subsampling the image background. This way, about 250.000 patches were collected for each class, whose examples are shown in Fig. 2.

However, for the training of RootNet a data augmentation step has been performed, where images have been randomly rotated around the center point, flipped (both horizontally and vertically), and adjusted in terms of sharpness, brightness, contrast, or saturation. All the augmentation operations do not affect the center point of the patch, so the patch does not change its class after being augmented. No color jitters have been introduced with the aim of avoiding the effect of introducing unlikely data in the dataset.

3.3. RootNet architecture

In the experiments, a straightforward but effective CNN-based architecture called *RootNet* is proposed, which is made of three stacked CNN layers with max pooling and ReLU activation. To design RootNet, the rules described by [16] have been followed, doubling the number of convolution filters when the feature map size is halved. Afterward, a fully connected layer was used, followed by a sigmoid activation function. The sigmoid activation function is chosen over the softmax activation function as the problem is framed as a *binary classification* task, where the network is required to establish whether each patch is either a positive or a negative sample. As for the loss function, binary cross-entropy has been used, with SGD as the optimization algorithm. A summary of the architecture of RootNet is shown in Fig. 3.

3.4. RootNet evaluation

As already explained in Section 3.3, our experiment aims to distinguish between *positive* (i.e., *roots*) and *negative* (i.e., *no-roots*) patches. Hence, the output feature map of the latest convolution layer (Conv3 in Fig. 3) is provided to a fully connected layer with a sigmoid activation function, whose output is in the range $[0, 1]$. Hence, given a threshold σ, the model provides a positive outcome if the prediction for the ith image is above σ, and a negative outcome otherwise.

The value of σ directly influences four scores that can be used to evaluate the performance of RootNet, specifically:

- **True Positives** (**TP**), that is, the number of positive samples that have been correctly classified.
- **True Negatives** (**TN**), that is, the number of negative samples that have been correctly classified.
- **False Positives** (**FP**), that is, the number of negative samples that have been incorrectly classified as positive samples.
- **False Negatives** (**FN**), that is, the number of positive samples that have been incorrectly classified as negative samples.

Adjusting the threshold via threshold-moving is a simple yet effective technique to improve classification performance [17]. To explain this, let us briefly recall the definitions of *recall* and *precision*:

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

In other words, the model achieves higher precision when $FP \to 0$, while it achieves higher recall when $FN \to 0$. However, in the case of binary classification, the value for the different scores is related to the σ value. A higher σ value implies that the model will misclassify fewer negative samples, reducing the overall FP. At the same time, however, the model will also misclassify a higher number of positive samples, leading to higher FN. Consequently, these combined effects will lead to higher precision and lower recall. On the other hand, a lower value for σ causes the opposite effect, reducing precision, but improving recall.

Hence, as the aim is to experimentally choose a value for σ to optimize both precision and recall, the *F1 score* metric is used, that embeds both precision and recall in its formulation:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (4)$$

4. Experimental results

In this section, the results achieved by the proposed method on the dataset described in Section 3.1 are described.

4.1. RootNet performance

First, the results of the RootNet architecture have been evaluated by varying the input image size in terms of precision, recall, accuracy, and F1 score. Specifically, three models with three different image input sizes have been used, that is, 257×257 (i.e., *RootNet-257*), 129×129 (i.e., *RootNet-129*), and 65×65 (i.e., *RootNet-65*). From these networks, the raw value predicted by the binary classifier has been extracted, that is, the raw value extracted by the sigmoid activation function. Several fixed values for the σ threshold are used to compute evaluation metrics. The results are reported in Figs. 4(a), 4(b) and 4(c). Furthermore, the numerical values achieved by the metrics at the threshold of $\sigma \sim 0.45$ are shown in Table 1.

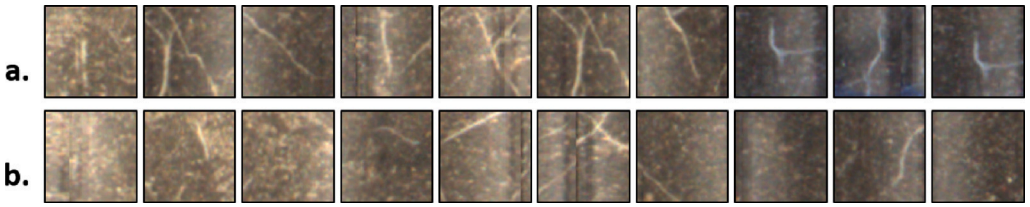


Fig. 2. RootNet dataset patches arranged in two rows: row a. that shows examples from the root class and row b. that shows examples from the non-root class. All the patches must span the highest number of background configurations possible to consider root image complexity. A non-root patch can have roots in the surroundings, but the center point must be a non-root.

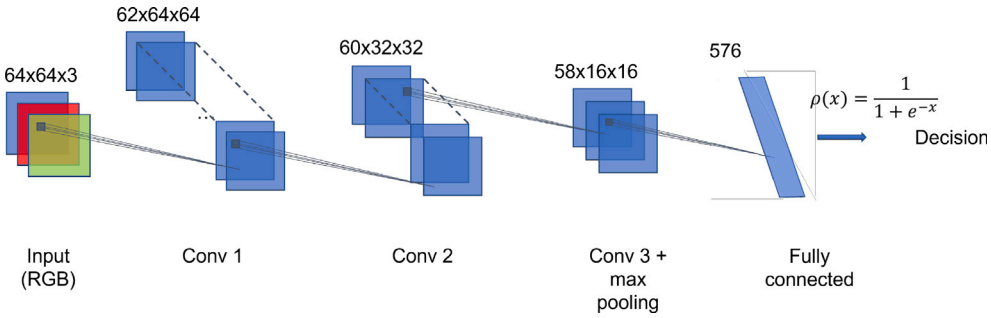


Fig. 3. RootNet architecture. The proposed architecture sends the RGB image through three different convolutional layers, with a decreasing density of the applied kernels. After the third convolution, a max pooling layer is applied to retain relevant features, which are then fed to a fully connected layer and, finally, to the decision layer.

Table 1
Metrics achieved by RootNet at a fixed value of $\sigma = 0.45$ after data augmentation.

Model	A (%)	P (%)	R (%)	F1 (%)
RootNet-257	**92.47**%	**91.97**%	92.71%	**92.34**%
RootNet-129	92.36%	91.65%	92.96%	92.30%
RootNet-65	92.22%	91.38%	**93.00**%	92.18%

Table 2
Metrics achieved by RootNet at a fixed value of $\sigma = 0.45$ without augmentation.

Model	A (%)	P (%)	R (%)	F1 (%)
RootNet-257	86.40%	**98.10**%	73.62%	84.12%
RootNet-129	86.18%	97.98%	73.44%	83.96%
RootNet-65	**87.81**%	97.95%	**76.89**%	**86.15**%

Table 2 shows the classification results achieved by the available configurations of RootNet without data augmentation. The results clearly show how data augmentation improves the overall performance of the networks. Interestingly, this is mainly related to a lower recall achieved by the network when trained on non-augmented data, resulting in a decrement in the performance of the network in the correct identification of root patches. This could be ascribed to the augmentation steps that keep the central point belonging to a root (or no-root), that increase the variability of the observed scene, resulting in better performance.

The first thing to notice is that the precision value shows direct proportionality with the σ threshold. In contrast, the recall shows inverse proportionality to the same threshold. As a consequence of this behavior, the accuracy and F1 curves show an inverted U-shape. This result is stable for the three RootNet models, regardless of the input size, even if the numerical values slightly differ for each architecture. From the analysis of these curves, it is possible to define proper σ threshold values to analyze the results produced by RootNet. For example, fixing a desired precision and recall values of 0.95, it can be defined $\sigma_p \sim 0.6$ and $\sigma_r \sim 0.3$ to filter an image processed by RootNet with the following logic:

- If a pixel has a predicted outcome value greater than or equal to σ_p, it is labeled as *root*.
- If a pixel has a predicted outcome value less than or equal to σ_r, it is labeled as *background*.
- Otherwise, it is labeled as *unknown*.

With reference to Fig. 4, the same logic can be applied using a single threshold, obtained when $\sigma_p = \sigma_r = \arg\max_{\sigma}(F1) \sim 0.45$, as the higher values of the F1 score are achieved for a threshold value between 0.4 and 0.5. These threshold values will be used in the next experiment to provide a qualitative evaluation of the images processed by RootNet, hence achieving a comparison with the SegRoot model on our dataset.

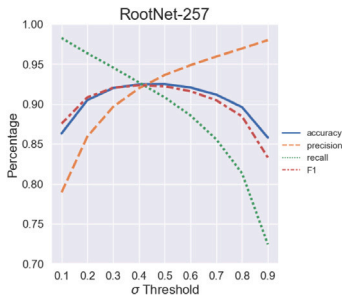
4.2. Comparison with SegRoot

This section compares the results of RootNet with those of SegRoot. However, it must be considered that the two networks use different underlying principles: a U-shaped encoder/decoder network for SegRoot and a stacked set of Convolutional-ReLU-Max pooling layers with a binary classifier on top of them for RootNet. Hence, a direct, quantitative comparison of classic metrics (e.g., accuracy) may not be suited for the task. Consequently, a qualitative evaluation is proposed comparing the original image, the ground truth, and the results achieved by both networks.

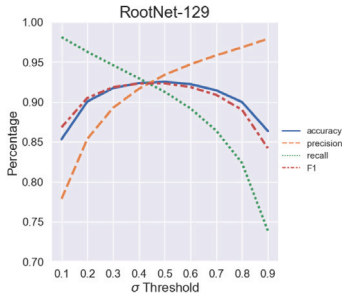
In Fig. 5, a visual comparison of the results achieved by SegRoot and the three different versions of RootNet is shown.

From Fig. 5(d), it can be seen that SegRoot provides the best results when used with the original weights, as retraining it on our dataset introduces a significant quantity of noise. However, by comparing the results achieved by SegRoot with the ground truth, it can be seen that it cannot capture the finer details, such as the smaller parts of the RSA. Furthermore, SegRoot misclassifies some of the artifacts introduced by the merging procedure applied on the original image (cfr. Section 3.1) as parts of the RSA. As for our architecture, the qualitative comparison shows that both RootNet-65 (Fig. 5(e)) and RootNet-129 (Fig. 5(f)) successfully capture fine-grained details about the RSA, with RootNet-129 achieving less noise in the bottom of the image, where no roots are available.

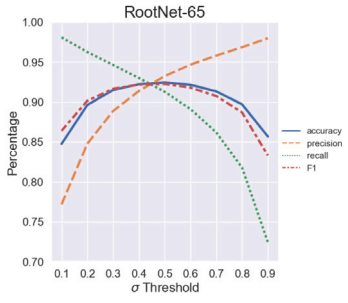
To further extend our comparison, let us qualitatively evaluate Fig. 6, where the results achieved by SegRoot are compared with the ones achieved with RootNet-65 on four different RSAs, selected according to both the density and the length of available roots. Specifically:



(a) RootNet-257

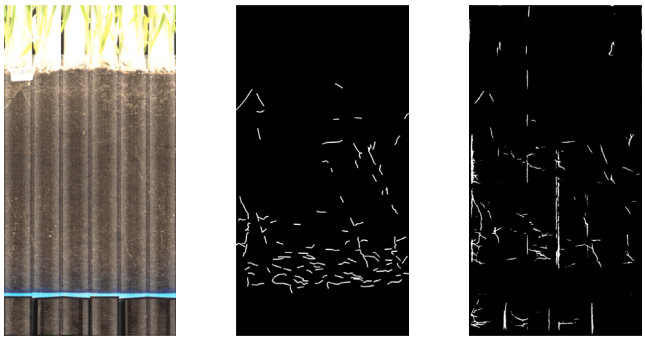


(b) RootNet-129

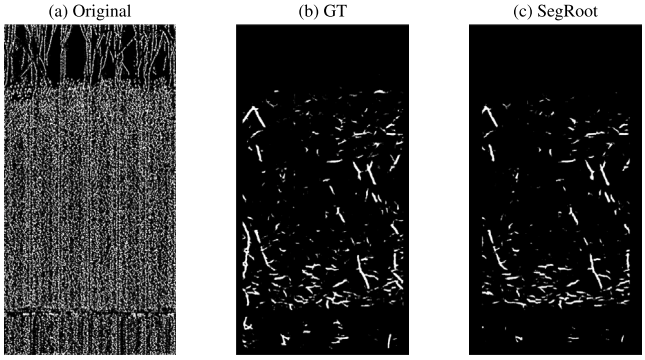


(c) RootNet-65

Fig. 4. From left to right, evaluation of Accuracy, Precision, Recall, and F1-score for RootNet-257, RootNet-129, and RootNet-65 at σ threshold levels from 0.1 to 0.9, sampled with a step of 0.1.



(a) Original (b) GT (c) SegRoot



(d) SegRoot - RT (e) RootNet-65 (f) RootNet-129

(g) RootNet-257

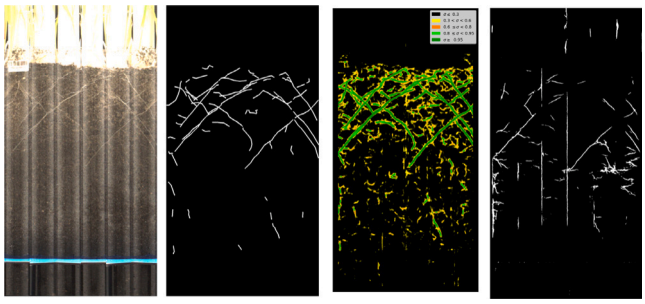
Fig. 5. Results achieved on a sample image. From left to right: the original image (5a), the ground truth (5b) manually extracted by domain experts, the results achieved by SegNet with its original weights (5c) and after being retrained on our dataset (5d), and the results achieved by RootNet-65 (5e), RootNet-129 (5f), and RootNet-257 (5g), respectively.

- **RSA 1** (*top left*) has been selected since a high density of long roots is visible in the upper part of the image.
- **RSA 2** (*top right*) has been selected since there is a high density of both long and short roots over the whole image.
- **RSA 3** (*bottom left*) has been selected due to the low density of the visible short roots.
- **RSA 4** (*bottom right*) has been selected due to the high density of short roots in the bottom part of the image.

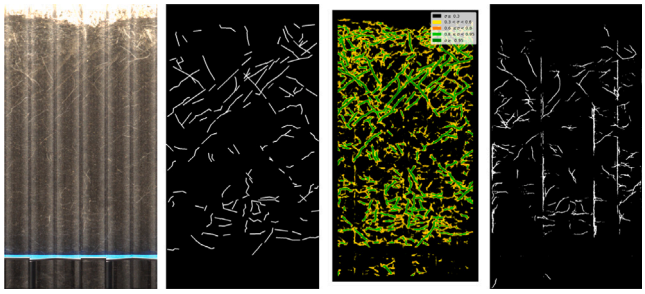
In each subfigure of Fig. 6 is shown, from left to right, the original image, the ground truth, the results achieved using RootNet-65, and the results achieved by SegRoot with its original weights. Specifically, the results provided by RootNet-65 are described in terms of the values of *sigma* for each pixel. Hence:

- If the pixel is colored in dark green, the network has classified it as a root with a confidence score above 0.95.
- If the pixel is colored in green, the network has classified it as a root with a confidence score between 0.8 and 0.95.
- If the pixel is colored in orange, the network has classified it as a root with a confidence score between 0.6 and 0.8.
- If the pixel is colored in yellow, the network has classified it as a root with a confidence score between 0.3 and 0.6.
- If the pixel is colored in black, the network has classified it as a root with a confidence score below 0.3.

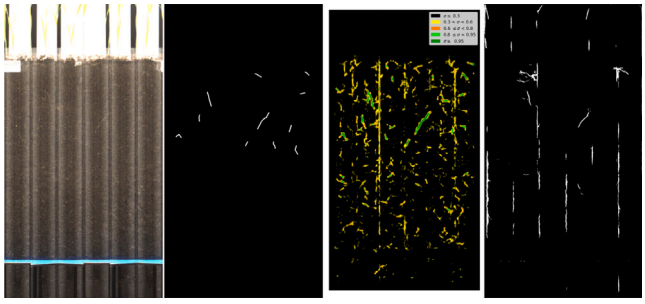
In other words, according to the two-thresholds formulation described in Section 4.1, pixels colored in orange, green, and dark green can be considered roots with a high confidence level. On the other hand, pixels colored in black can be considered part of the background. Finally, pixels colored in yellow are labeled as "uncertain" and, as it can be seen, mostly belong to the zones relative to the artifacts introduced by the preprocessing on the images or to the zones surrounding the roots. This is also desirable, as it can highlight root parts that are effectively within the RSA but have not been labeled by the domain expert as too dim in their appearance on the image. As seen from the images, RootNet outperforms SegRoot in the cases shown in Figs. 6(a), 6(b), and 6(d), which account for dense zones of long and short roots, providing high reliability, especially by considering the two-thresholds formulation proposed. As for the case shown in Fig. 6(c), RootNet appears to be able to correctly characterize roots, which are also available in the ground truth; however, several points also appear with values of the confidence score above 0.6, which can be taken back in part to the artifacts within the image and in part to several dim structures within



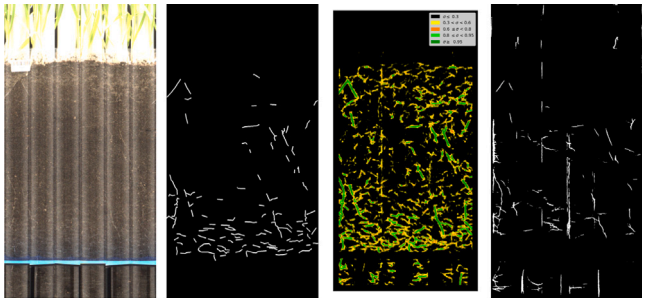
(a) RSA 1 - high density of long roots.



(b) RSA 2 - high density of long and short roots.



(c) RSA 3 - low density of roots.



(d) RSA 4 - High density of short roots.

Fig. 6. Qualitative comparison between SegRoot with original weights and RootNet-65. From left to right, respectively, the original image, the ground truth, RootNet-65 results, and finally SegRoot binary mask are reported.

the RSA. Therefore, in this case, using the single-threshold formulation may be preferable.

4.3. Quantitative comparison

To further assess the performance of RootNet, a pixel-based quantitative comparison against SegRoot was proposed. Specifically, four metrics were used: precision, recall, F1 score and Hausdorff distance between the ground truth and a binary mask produced by each method. Results are shown in Table 3.

Table 3
Quantitative pixel-based comparison of RootNet against SegRoot.

Network	F1	P	R
SegRoot	11.58%	9.50%	20.42%
RootNet-65	17.56%	10.07%	**77.02**%
RootNet-129	**22.65**%	**13.77**%	67.78%
RootNet-257	21.40%	13.39%	60.62%

Table 4
Quantitative comparison of RootNet against SegRoot over patches of 3×3 pixels.

Network	F1	P	R
SegRoot	11.66%	9.60%	20.32%
RootNet-65	18.39%	10.60%	**76.68**%
RootNet-129	**23.62**%	**14.48**%	67.70%
RootNet-257	22.41%	14.16%	60.40%

Table 5
Quantitative comparison of RootNet considering the border effect.

Network	F1	P	R
RootNet-65	17.64%	10.06%	**79.33**%
RootNet-129	**23.07**%	**13.71**%	74.71%
RootNet-257	22.40%	13.39%	74.25%

It is important to underline that, as already stated in Section 4.2, a direct comparison in terms of standard metrics among these models is not straightforward, as they are based on different considerations and working principles. In particular, the problem solved by RootNet is intrinsically formulated as a probability estimation, therefore the informative content output by the proposed method is not a simple binary mask. Moreover, to frame the problem and prepare the dataset, particular attention was payed to the ground truth labeling, privileging thin lines that certainly highlight a root in the images, due to the major voting procedure described before. As such, even if the quantitative comparison is based on a pixel-level evaluation of the results achieved using the networks in inference mode over 17 validation images, considering such binary masks and the ground truth labeled this way could lead to relatively low values of the F1 score. For this reason, in order to better evaluate the performance of the models in the most unbiased way, we also provided a computation of the Hausdorff distance between the ground truth and all the binary masks obtained by the networks. Table 3 shows that RootNet-129 outperforms the other models in terms of F1 score and precision, while RootNet-65 achieves the highest value for recall. Still, these values must be taken in the context of a pixel-based evaluation, which can be inherently biased by minimal offset errors in the prediction. In other words, a displacement of the prediction performed by the network of a negligible number of pixels, either vertically or horizontally, can significantly impact the values provided by the metrics. As such, the results were also validated considering the average prediction of a patch of 3×3 pixels. The results are shown in Table 4, and confirm the ones already achieved in Table 3.

Finally, let us consider the border effect introduced by RootNet when used in inference. In fact, during the proposed tests, the model was used in inference without introducing any extra padding effect to avoid repetition bias. However, this imposes a tradeoff in that the outermost $\frac{N-1}{2}$ pixels will not be considered during the analysis, with N the patch size RootNet considers during training. Consequently, accounting for these border effects yields the results shown in Table 5.

Interestingly, when the border effect is considered, the precision is slightly affected, along with the overall F1 score, but the recall is noticeably improved. This is mainly related to the fact that the border

Table 6
Quantitative comparison of the Hausdorff distance between the ground truth and the binary masks computed by the network models.

Network	Hausdorff distance
SegRoot	24.61
RootNet-65	**1.78**
RootNet-129	3.27
RootNet-257	3.05

pixels are not predicted as belonging to roots, hence the overall number of false negatives decreases, therefore improving the recall achievable by the network. Finally, in Table 6 the Hausdorff distance between the ground truth and the validation binary masks is reported, showing that the proposed approach is able to provide a root mask with a distance error of less than 2 pixels in the best case and less than 4 pixels in the worst case.

5. Conclusions and future works

This work has proposed an alternative formulation of the RSA segmentation problem that does not require a U-shaped network but relies on binary classification via probability map estimation to classify pixels of the original image as roots or background. This approach has provided optimal quantitative results regarding the accuracy and the F1-score using small CNNs with only three stacked convolutional layers. This CNN model was specifically selected to test the effectiveness of the end-to-end pipeline. In this sense, future works will be directed towards testing more complex architectures, aiming at finding the optimal trade-off between achieved accuracy and computational load. As it provides adequate performance with small models, it has an overall reduced computational cost compared to other approaches requiring more resource-exhaustive architectures. Furthermore, the approach is flexible, as the value of the optimal thresholds used for the final classification can be tuned to achieve the desired quantitative metrics.

The practicability and feasibility of this approach are especially relevant in scenarios where there is a lack of labeled data. Labeling RSAs is costly for the domain expert. Still, the proposed approach can generate many patches from a relatively limited number of images, providing an effective tool to identify complex RSAs with high reliability. Furthermore, the approach can be easily scaled, and an optimized version can exploit the parallel computational capabilities of GPUs to achieve quasi-real-time performance over large amounts of data.

Future works will be focused on exploring different architectures, which can also include different types of layers, to improve both the overall metric and the efficiency of the network, as well as testing the proposed model on images of different plant species. Furthermore, new evaluation metrics will be provided to quantitatively compare the results achieved by RootNet with other networks in a simple and unbiased fashion.

Data availability

Data will be made available by the researchers upon specific request. The code is available online on GitHub at the following address: https://github.com/vitoreno/pyroots.git

CRediT authorship contribution statement

A. Cardellicchio: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **F. Solimani:** Conceptualization, Data curation, Investigation, Validation, Writing – original draft. **G. Dimauro:** Formal analysis, Investigation, Supervision, Writing – original draft, Writing – review & editing. **S. Summerer:** Data curation, Resources, Validation, Visualization, Writing – original draft. **V. Renò:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available by the researchers upon specific request. The code is available online on GitHub at the following address: https://github.com/vitoreno/pyroots.git.

Acknowledgments

The activities described in this work are within the research projects *PHENO – Accordo di collaborazione tra ALSIA e CNR STIIMA – ref. prot. CNR STIIMA 3621/2020 and E-Crops – Technology for Sustainable Digital Agriculture*. The authors would like to thank Mr. Michele Attolico for technical support.

References

[1] T. Galkovskyi, Y. Mileyko, A. Bucksch, B. Moore, O. Symonova, C.A. Price, C.N. Topp, A.S. Iyer-Pascuzzi, P.R. Zurek, S. Fang, J. Harer, P.N. Benfey, J.S. Weitz, GiA Roots: Software for the high throughput analysis of plant root system architecture, BMC Plant Biol. 12 (1) (2012) 116, http://dx.doi.org/10.1186/1471-2229-12-116.

[2] V. Renò, M. Nitti, P. Dibari, S. Summerer, A. Petrozza, F. Cellini, G. Dimauro, R. Maglietta, Automatic stitching and segmentation of roots images for the generation of labelled deep learning-ready data, in: Multimodal Sensing and Artificial Intelligence: Technologies and Applications II, vol. 11785, SPIE, 2021, pp. 174–179, http://dx.doi.org/10.1117/12.2595062.

[3] S.J. Mooney, T.P. Pridmore, J. Helliwell, M.J. Bennett, Developing X-ray computed tomography to non-invasively image 3-D root systems architecture in soil, Plant Soil 352 (1) (2012) 1–22, http://dx.doi.org/10.1007/s11104-011-1039-9.

[4] C. Planchamp, D. Balmer, A. Hund, B. Mauch-Mani, A soil-free root observation system for the study of root-microorganism interactions in maize, Plant Soil 367 (1) (2013) 605–614, http://dx.doi.org/10.1007/s11104-012-1497-8.

[5] J.S. Perret, M.E. Al-Belushi, M. Deadman, Non-destructive visualization and quantification of roots using computed tomography, Soil Biol. Biochem. 39 (2) (2007) 391–399, http://dx.doi.org/10.1016/j.soilbio.2006.07.018.

[6] R. Rellán-Álvarez, G. Lobet, H. Lindner, P.-L. Pradier, J. Sebastian, M.-C. Yee, Y. Geng, C. Trontin, T. LaRue, A. Schrager-Lavelle, C.H. Haney, R. Nieu, J. Maloof, J.P. Vogel, J.R. Dinneny, GLO-Roots: An imaging platform enabling multidimensional characterization of soil-grown root systems, eLife 4 (2015) e07597, http://dx.doi.org/10.7554/eLife.07597, publisher: eLife Sciences Publications, Ltd.

[7] P. Borianne, G. Subsol, F. Fallavier, A. Dardou, A. Audebert, GT-RootS: An integrated software for automated root system measurement from high-throughput phenotyping platform images, Comput. Electron. Agric. 150 (2018) 328–342, http://dx.doi.org/10.1016/j.compag.2018.05.003.

[8] N. Narisetti, M. Henke, C. Seiler, R. Shi, A. Junker, T. Altmann, E. Gladilin, Semi-automated root image analysis (saRIA), Sci. Rep. 9 (1) (2019) 19674, http://dx.doi.org/10.1038/s41598-019-55876-3, number: 1 Publisher: Nature Publishing Group.

[9] T. Wang, M. Rostamza, Z. Song, L. Wang, G. McNickle, A.S. Iyer-Pascuzzi, Z. Qiu, J. Jin, SegRoot: A high throughput segmentation method for root image analysis, Comput. Electron. Agric. 162 (2019) 845–854, http://dx.doi.org/10.1016/j.compag.2019.05.017.

[10] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, http://dx.doi.org/10.1109/TPAMI.2016.2644615, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[11] F. Milletari, N. Navab, S.-A. Ahmadi, V-Net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 Fourth International Conference on 3D Vision, 3DV, 2016, pp. 565–571, http://dx.doi.org/10.1109/3DV.2016.79.

[12] C. Shen, L. Liu, L. Zhu, J. Kang, N. Wang, L. Shao, High-throughput in situ root image segmentation based on the improved DeepLabv3+ method, Front. Plant Sci. 11 (2020).

[13] S. Teramoto, Y. Uga, A deep learning-based phenotypic analysis of rice root distribution from field images, Plant Phenomics 2020 (2020) 1–10, http://dx.doi.org/10.34133/2020/3194308.

[14] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin, B. Hoff, TOsmanov, D. Kruchinin, A. Zankevich, DmitriySidnev, M. Markelov, Johannes222, M. Chenuet, a. andre, telenachos, A. Melnikov, J. Kim, L. Ilouz, N. Glazov, Priya4607, R. Tehrani, S. Jeong, V. Skubriev, S. Yonekura, vugia truong, zliang7, lizhming, T. Truong, Opencv/cvat: v1.1.0, 2020, http://dx.doi.org/10.5281/zenodo.4009388.

[15] Dataset management framework (datumaro), 2022, URL https://github.com/openvinotoolkit/datumaro.

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015, http://dx.doi.org/10.48550/arXiv.1512.03385, arXiv:1512.03385 [cs].

[17] F. Provost, Machine learning from imbalanced data sets 101, in: Proceedings of the AAAI'2000 Workshop on Imbalanced Data Sets, vol. 68, AAAI Press, 2000, pp. 1–3.