Computers and Electronics in Agriculture xxx (xxxx) 109457

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

A CNN-based framework for estimation of root length, diameter, and color from *in situ* minirhizotron images

Faina Khoroshevsky [a,*], Kaining Zhou [b,c], Aharon Bar-Hillel [a], Ofer Hadar [d], Shimon Rachmilevitch [c], Jhonathan E. Ephrath [c], Naftali Lazarovitch [c], Yael Edan [a]

[a] Department of Industrial Engineering and Management, Beer Sheva, Israel
[b] The Jacob Blaustein Center for Scientific Cooperation, The Jacob Blaustein Institutes for Desert Research, Sede Boqer, Israel
[c] The French Associates Institute for Agriculture and Biotechnology of Drylands, The Jacob Blaustein Institutes for Desert Research, Ben-Gurion University of the Negev, Sede Boqer, Israel
[d] Department of Communication Systems Engineering, School of Electrical and Computer Engineering, Beer Sheva, Israel

ARTICLE INFO

Keywords:
Root phenotyping
Minirhizotron
Convolutional neural network
Grapevine root dataset
Segmentation free

ABSTRACT

This work presents a framework based on convolutional neural networks (CNNs) to estimate root traits (length, diameter, and color) from minirhizotron (MR) imagery. The proposed framework uses a set of reusable sub-network modules to compose different networks for object (i.e., root) detection and attribute (i.e., trait) estimation for per-root and per-image root phenotyping tasks. It provides a solution without requiring root segmentation. The first step in per-root phenotyping involves detecting the roots in the image; the traits of each detected root are then estimated. Per-image root phenotyping estimates aggregated root trait values, including total root length (TRL), mean root diameter, and percentage of white root. *Regression-based* and objects' *points-detection-based* variations are demonstrated for both per-root and per-image root trait estimation. Five network architectures are presented, two of which were previously used for TRL estimation (and are now evaluated for estimating mean root diameter and white root percentage), and three of which are new.

The proposed framework is demonstrated on an annotated grapevine root dataset comprising 531 images, made publicly available as part of this paper. All images were acquired *in situ* using an MR system and annotated with Rootfly software. *Regression-based* modules used for individual detected roots yielded errors of 8.8%, 15.5%, and 23.5% for color, length, and diameter, respectively. The *points-detection-based* modules resulted in errors of 9.1%, 14.9%, and 25.0% for the same parameters. The image-level estimates showed errors of 11.5%–16.5% for white root percentage, 13.7%–16.0% for TRL, and 17.6%–22.1% for mean root diameter. We demonstrate that aggregating per-root estimations of diameter and color obtained with the new suggested architectures improves the per-image estimations of these traits relative to the direct per-image estimation that does not include per-root estimations. To demonstrate further the practicality of the suggested framework in deriving the vertical distribution of various root traits, an additional dataset of 132 root images from two different grapevine graft combinations was annotated (and also made publicly available as part of this paper). In this dataset, the per-image root traits were estimated for different soil depths and visually compared with human annotation results.

1. Introduction

A root's phenotype defines its physical and biochemical characteristics (Sood and Singh, 2021). These include traits such as surface area, length, volume, mean diameter, length density (Lupo et al., 2022), and color (Iversen et al., 2017).

A root system's architecture and growth vary greatly within and among species in response to changing growth conditions (Soda et al., 2017; Aidoo et al., 2018; Lupo et al., 2022). Knowledge of both is essential to understanding a plant system under diverse environmental conditions (Soda et al., 2017; Lupo et al., 2022). Root diameter affects a plant's functioning, and has been suggested as the main control of soil infiltration in semi-arid grasslands (Cui et al., 2019). It is also the char-

* Corresponding author.
E-mail addresses: bordezki@post.bgu.ac.il (F. Khoroshevsky), zhoy@post.bgu.ac.il (K. Zhou), aharon.barhillel@gmail.com (A. Bar-Hillel), hadar@bgu.ac.il (O. Hadar), rshimon@bgu.ac.il (S. Rachmilevitch), yoni@bgu.ac.il (J.E. Ephrath), lazarovi@bgu.ac.il (N. Lazarovitch), yael@bgu.ac.il (Y. Edan).

https://doi.org/10.1016/j.compag.2024.109457
Received 19 March 2024; Received in revised form 13 September 2024; Accepted 15 September 2024

acteristic that defines a root as being fine or coarse (McCormack et al., 2015; Prieto et al., 2016), with these categories differing in function and turnover rate (McCormack et al., 2015; Zhang and Wang, 2015). Root color is a common visual cue for researchers in estimating whether roots are aged or functionally active (e.g., absorbing or transporting water and nutrients; Iversen et al., 2017): very dark brown or black roots are considered dead, and white or light brown roots are considered alive (Hendrick and Pregitzer, 1992). A change in root color often indicates a change in root condition, such as lignification or late stage of senescence (Comas et al., 2000; Dannoura et al., 2008; Soda et al., 2017). When subjected to abiotic and biotic stresses (e.g., high salinity or worms), roots exhibit noticeable color differences from a control group (Soda et al., 2017; Song et al., 2023).

Examples of root phenotyping include a study of the response of olive roots to salt stress, whereby ion uptake and accumulation were quantified alongside measurements of root count, diameter, length, and age (color) (Soda et al., 2017); research of the mechanisms by which grapevine rootstocks adjust their structure and architecture when exposed to salt stress (Lupo et al., 2022); and quantification of the relationship between water uptake and traits such as the length and diameter of tomato roots in an effort to optimize irrigation frequency (Geng et al., 2023).

The main challenge of measuring roots arises from the destructive procedure involved, as plants must often be excavated for analysis (Wu et al., 2016; Liu et al., 2019; Lupo et al., 2022). Traditional methods such as profile wall measurement, block excavation, and soil cores (Pierret et al., 2005; Han et al., 2015) damage roots and distort their architecture, leading to inaccurate measurements and conclusions (Pierret et al., 2005). Washing roots can also destroy their original structure and cause entanglement of fragile fine roots, resulting in underestimated total root length (TRL) and inaccurate assessment of mean root diameter (Galdos et al., 2020). Alternatively, changing root dynamics in response to changing environmental conditions have been investigated using image-based root phenotyping techniques (Busener et al., 2020; Hui et al., 2022; Geng et al., 2023). Among these, the minirhizotron (MR) image acquisition system has been widely used for the repeated and non-destructive *in situ* study of roots (Rewald and Ephrath, 2013; Zhou et al., 2018). Important root traits (e.g., root color, diameter, and length) can be derived from MR imagery using image analysis software such as Rootfly (Wells and Birchfield, Clemson University, South Carolina, USA). However, this approach requires time-intensive manual annotation of the roots in each image to extract the root traits (Johnson et al., 2001; Danilevicz et al., 2021; Geng et al., 2023), which considerably limits the size and number of experiments that can be reasonably conducted.

Deep learning (DL) tools (Lecun et al., 2015) have been developed to automate the analysis of MR images, although reported results are limited to root length estimation and require image segmentation as a preliminary step (Bauer et al., 2022; Smith et al., 2022; Baykalov et al., 2023). Training a segmentation-based network (Yu et al., 2020; Gillert et al., 2021; Huang et al., 2023) necessitates substantial human involvement in annotating segmentation masks. This involves marking the contour of each root using an annotation tool to capture all the pertinent pixels of the roots in the image. The segmentation step, either through manual annotations (Geng et al., 2023) or the output of a trained segmentation network (Bauer et al., 2022; Smith et al., 2022; Baykalov et al., 2023), is followed by converting pixel data to actual root trait values (Bauer et al., 2022; Smith et al., 2022; Baykalov et al., 2023; Geng et al., 2023).

To our knowledge, apart from our previous study (Khoroshevsky et al., 2024) on estimating total root length (TRL, calculated as the sum of root lengths per image), there are no DL tools available that can automatically analyze root traits from MR images without segmentation. Additionally, we did not find prior results comparing DL tools for root diameter or color estimation with true values based on manual annotations or measurements.

The segmentation-based methods previously suggested for TRL estimation (Bauer et al., 2022; Smith et al., 2022; Baykalov et al., 2023) estimate TRL by skeletonizing the segmentation masks. Smith et al. (2022) presented the software RootPainter for image segmentation, based on a version of U-Net (Ronneberger et al., 2015). After applying RootPainter, root length estimates can be extracted by skeletonization of the generated segmentations and pixel counting. Bauer et al. (2022) also incorporated a pipeline for TRL estimation that combined root segmentation using RootPainter, whereas the TRL estimates were extracted as the sum of the Euclidean distances between the connected skeletal pixels of the root topology using RhizoVision Explorer software (Seethepalli et al., 2021). Baykalov et al. (2023) followed the approach of Smith et al. (2022) by estimating TRL based on segmentation skeletonization but developed an improved segmentation method using the U-Net architecture with pre-trained backbones as encoders.

This study presents a framework based on convolutional neural networks (CNNs; LeCun et al., 1998) to automatically estimate root length, diameter, and color for *in situ* root phenotyping from MR images. The suggested networks are composed of different modules, some of which are off-the-shelf and others developed in this research. The proposed new networks do not require a pre-segmentation step. Training requires only inherited information provided by the image analysis software Rootfly.

Using the suggested framework, five network architectures are composed to handle two types of root phenotyping tasks. The first is ***per-root estimation***, which involves detecting individual roots in an image and estimating for each detected root its length, diameter, and color (when roots are considered to be either white or brown). The second type is ***per-image phenotyping***, whereby the TRL, mean root diameter, and percentage of white roots are estimated for a whole image. We previously used two of the five architectures for per-image phenotyping for TRL estimation (Khoroshevsky et al., 2024). The current study further tests both architectures for additional per-image phenotyping (mean diameter and percentage of white roots) and suggests three additional architectures compiled using the suggested set of modules.

The framework is demonstrated on a dataset of 663 MR grapevine root images, made publicly available[1] as part of this research. The dataset consists of MR images acquired *in situ* and non-destructively with thorough annotations specially made as part of the current research. This dataset can be used for benchmarking future developments. Such datasets are important to enable comparison of different approaches and to the construction of generalized models (Farjon et al., 2023).

The remainder of this manuscript is organized as follows. Section 2 describes the methods including the framework. The results and a discussion of the different networks for the different phenotyping tasks are presented in Section 3. Finally, the conclusions are provided in Section 4.

2. HYPERLINK “SPS:id::Sec1” Materials and methods

2.1. Overall framework

The proposed framework is fitted with a CNN-based network for a specific task; this single network is composed of a given inventory of CNN-based modules (sub-networks). Section 2.2 outlines the proposed modules and their use for the different root phenotyping tasks. Appendix A describes the architectures in detail. We use these modules for two main computer vision tasks: object (root) detection and attribute (trait) estimation.

[1] The dataset is publicly available in the Zenodo repository [https://zenodo.org/record/8084106].

Objects can be detected by finding their bounding box (designated *bbox*) or by detecting the locations of specific points on them. The output is a density map (or heat map) showing the probability of each location containing a point.

Attribute estimation refers to estimating a single binary or continuous value for an attribute that applies to either a single object in an image or some kind of aggregate (e.g., sum or mean) for an image with multiple objects. Attributes can be estimated on a ***per-image*** or ***per-object*** basis, in each case employing different networks from the presented modules (Section 2.3). In this study, the *per-object* attribute estimation tasks involve estimating, for each root, continuous values for length and diameter and a binary value for color (1 for white or 0 for dark brown). These estimations are based on first detecting the roots in an image, and then estimating the trait values for each detected root. The *per-image* attribute estimations conducted here are assessed using the TRL, mean diameter, and percentage of white roots, based on all roots in an image.

2.2. HYPERLINK "SPS:id::Sec2" Proposed modules

2.2.1. Modules for detection tasks

The following modules are used for the detection tasks (Fig. 2-1).

The combination of the ***"Backbone"***, ***"Find (for bbox detection)"***, and ***"Where"*** modules implement the (off-the-shelf) object detection network RetinaNet (Lin et al., 2017b) from which we can estimate bounding boxes (Fig. 2-1a). The combination of the ***"Backbone"*** and ***"Find (for points detection)"*** modules implements a *points-detection-based* network, which outputs a heat map for the object detection network (Fig. 2-1b).

***"Backbone"* module (**Appendix A1**):** This module creates five feature-rich representations (termed $P_3 - P_7$) of the original input image, where the spatial dimensions of P_i are $2^{-i} \cdot (h, w)$ for the original image size (h, w). It applies ResNet-50 (He et al., 2016) as a dense CNN with a Feature Pyramid Network (FPN) (Lin et al., 2017a) on top of it.

***"Find"* module (**Appendix A2**):** This module is also a CNN; it has some internal variations that depend on its specific use. It is trained to generate heat maps (one or several) that spatially locate point presence in an input tensor representation (a feature representation received from the *"Backbone"* module). It is also used for attribute estimation by outputting additional features to be used in the following *detection and regression* (designated *"D+R"*) module (Section 2.2.2) that performs detection-based attribute estimations. The *"Find"* module is used in two main output modes, as follows.

1. ***bbox detection*** – The *"Find"* module is used as a component of a network that detects bounding boxes, re-implementing the classification sub-network of the RetinaNet architecture. When estimating the bounding box, it estimates each object's center point presence probability in the range [0,1] with nine output maps per object class corresponding to the RetinaNet output.
2. ***points detection*** – This mode of the *"Find"* module is applied to estimating attributes (length, diameter, and color) based on the points detection. It is used for a general estimation of the probability (in range [0,1]) of the locations of points, outputting a single heat map.

***"Where"* module (**Appendix A3**):** This module implements the bounding box (*bbox*) regression sub-network of RetinaNet and is used in composing networks for object detection through bounding box localization. For each input tensor, it estimates for every position and possible anchor (among the nine considered) a four-dimensional bounding box refinement vector (total of 36 values).

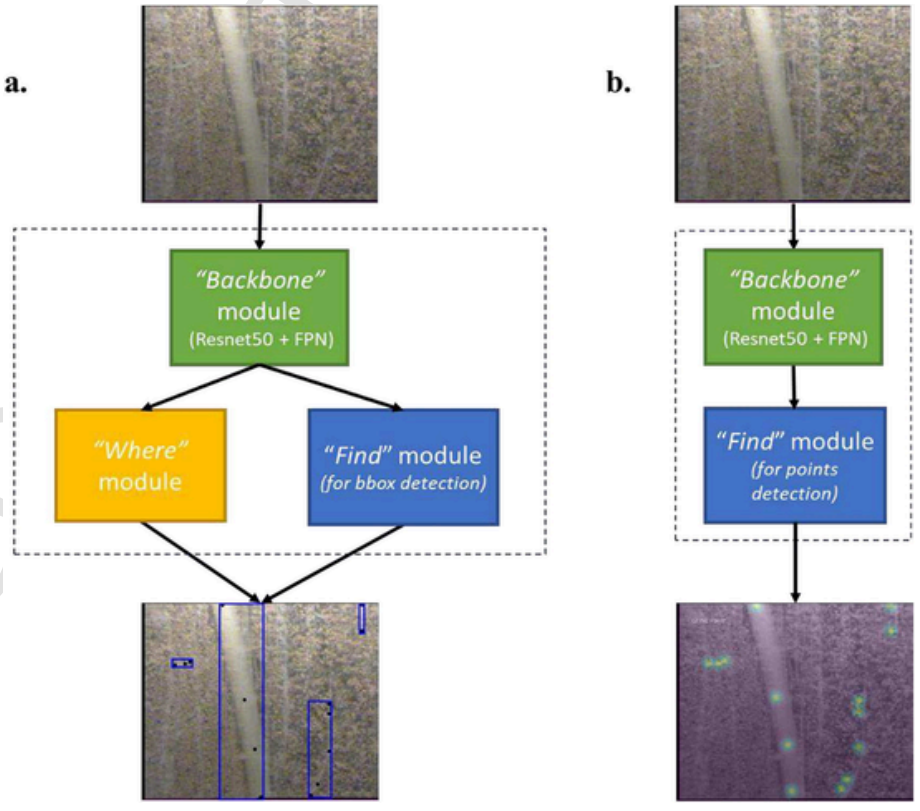

Fig. 2–1. Composition of modules for object detection, with the expected output of each variation based on **a.** bounding box detection and **b.** points-detection-based heat map generation.

2.2.2. HYPERLINK "SPS:id::Sec3" Modules for attribute estimation

The following modules are used for attribute estimation. For any RGB input, an attribute can be estimated by either a combination of the ***"Backbone"*** module with a multiple-scale *regression-based* (designated ***"MSR"***) module that creates a network for *regression-based* attribute estimation, or a combination of ***"Backbone"***, ***"Find (for points detection)"***, and a ***"D+R"***-based (designated ***"D+R"***) module that creates a network for *points-detection-based* attribute estimation.

"MSR" module **(**Appendix A5**):** This module is relevant to *regression-based* attribute estimations. It is a CNN that receives as input a representation tensor (one of the five-scale representations from the *"Backbone"* module) and outputs a single estimation. It has two different modes: *"MSR (for binary)"* for binary output and *"MSR (for continuous)"* for continuous output values.

"D+R" **module (**Appendix A6**):** This module can generate any *points-detection-based* estimation (continuous or binary) based on its input heat map from the *"Find"* module and additional useful features extracted from the outputs of the *"Find"* module in the *"Find (for points detection)"* variation. This module is relevant to *points-detection-based* estimations, and it follows the *"Find (for points detection)"* module, from which it receives features for attribute estimation. The output modes, *"D+R (for binary)"* and *"D+R (for continuous)"*, correspond to binary and continuous outputs, respectively.

2.3. HYPERLINK "SPS:id::Sec4" Composing the networks

2.3.1. Per-root trait estimation

Per-root estimation is performed by 1) root detection with the *"Backbone"*, *"Find (for bbox detection)"*, and *"Where"* modules; 2) cropping the roots from the original image and resizing them with the *"RoI-Align"* module (He et al., 2017; Appendix A4); and 3) passing each image crop to the 'phenotyping' section of the network that outputs length, diameter, and color estimations per-root. The phenotyping section is composed of an additional *"Backbone"* module followed by trait estimation modules with two possible variations (Fig. 2-2).

The first option is relevant to the *points-detection-based* network variation, which uses a combination of a *"Find (for points detection)"* module with three *"D+R"* modules: two (for length and diameter estimation) being *"D+R (for continuous)"* modules, and one *"D+R (for binary)"* module for color estimation (Fig. 2-2a). The second option, which is relevant to *regression-based* attribute estimation, involves a network composed of the *"Backbone"* module followed by three *"MSR"* modules: two (for length and diameter estimations) being *"MSR (for continuous)"* modules, and one *"MSR (for binary)"* for color estimation (Fig. 2-2b).

For both options (Fig. 2-2), each of the two main sections of the suggested networks (detection and 'phenotyping') has its own loss functions, and we do not propagate the gradient through the *"RoI-Align"* module. The training was conducted first for the detection section. Its weights were frozen, and only then was the training of the other modules started. This allows isolation of the performance of the sub-networks that perform the estimation of the length, diameter, and color, from the detector performance. The architectures were compared using the same detector (i.e., the fixed detection section).

2.3.2. HYPERLINK "SPS:id::Sec5" Network compositions from pre-trained modules

An additional network (Fig. 2-3) for per-root phenotyping, following the *points-detection-based* approach, was composed of pre-trained modules. The network was composed of the object detection section with an "*RoI-Align*" module and two branches from the "*RoI-Align*" module in the 'phenotyping' section: the left branch is for length estimation and the right branch is for diameter and color estimation.

This network is for inference only (estimations from new data); i.e., it is meant to be used as-is during testing, without additional training. By composing a network from pre-trained modules, we can use a computationally heavy network (two *"Backbone"* modules in the 'phenotyping' section in addition to a *"Backbone"* in the detection section) without the need to train it from scratch. This network is compared to the previous *points-detection-based* network (Fig. 2-2a) in Section 3.1.2.

2.3.3. Per-image trait estimation

We previously used direct estimation networks to estimate the TRLs of other root crops (Khoroshevsky et al., 2024, Fig. 2-4); they are applied here to estimate the mean diameter and percentage of white roots in addition to TRL.

Image-level estimations are performed with either a *regression-based* network composed of a *"Backbone"* and the relevant variation of an *"MSR"* module (Fig. 2-4a) or with a *points-detection-based* network composed of a *"Backbone"*, *"Find"*, and the relevant variation of a *"D+R"* module (Fig. 2-4b). The input to each network is the original RGB image.

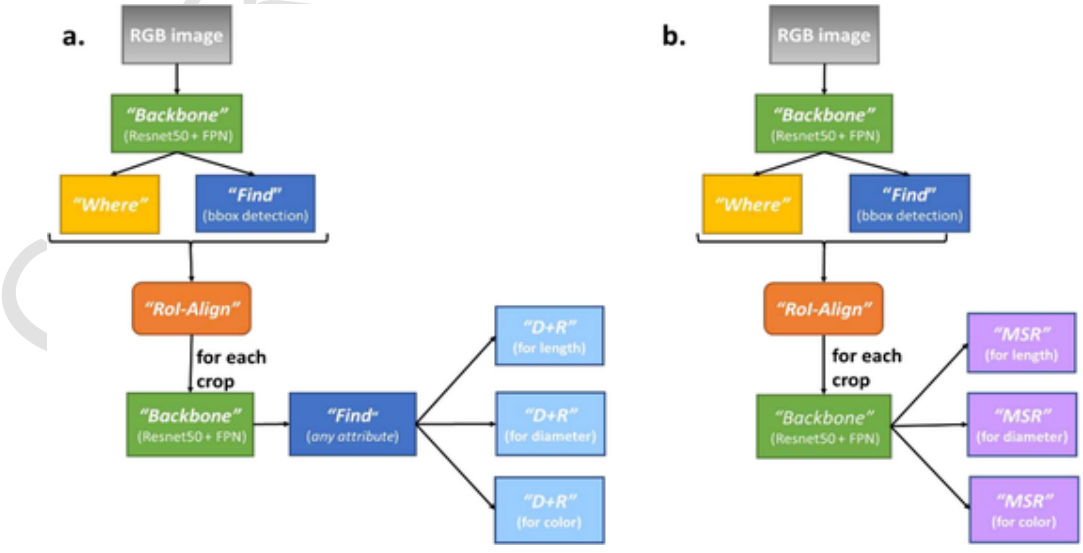

Fig. 2–2. Per-root trait estimation (length, diameter, and color) with two network variations based on root detection followed by **a.** a points-detection-based network or **b.** a regression-based network.

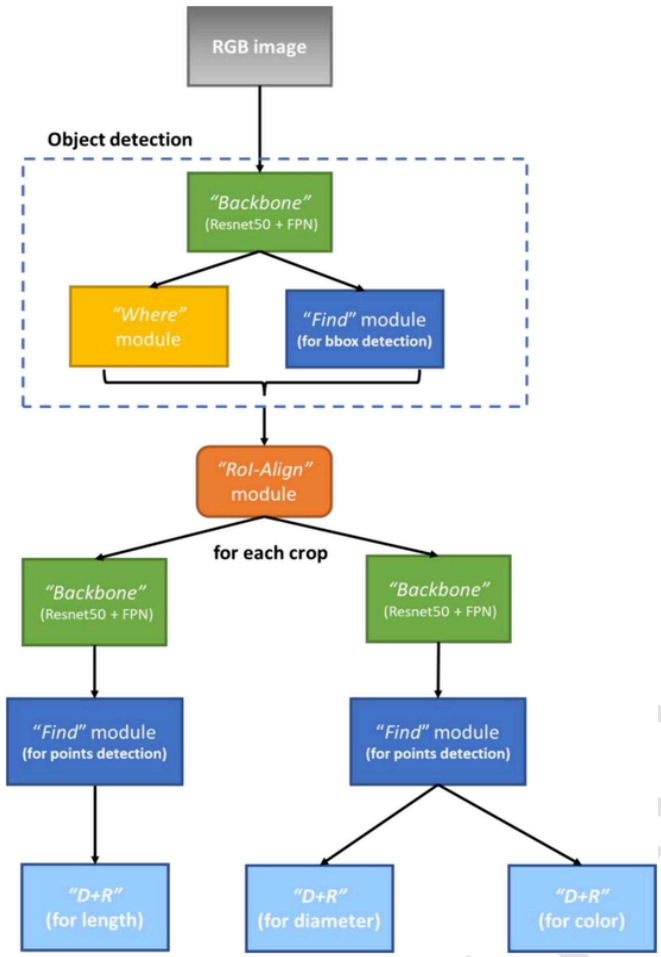

Fig. 2–3. Per-root trait estimation (length, diameter, and color) with points-detection-based network architecture composed of pre-trained modules that have two "Backbone" and two "Find" modules in the 'phenotyping' section.

Per-image traits can also be estimated by aggregating the per-root trait estimations for the detected roots in an image. The TRL in an image is calculated as the sum of the root lengths estimated for each detected root. The mean diameter and percentage of white roots are calculated as the average of the estimated per-root diameters and the average of the estimated (binary) per-root color in the image, respectively. These two approaches for per-image trait estimates are compared in Section 3.2.

2.4. Training the networks with loss balancing

Loss balancing for per-root estimations. Since the different phenotyping modules (*"D+R"* or *"MSR"*) estimate length, diameter, and color, their outputs are on different scales. The color is a binary value with an estimation error in the range of 0–1, whereas the length and diameter modules estimate continuous values on different scales. Therefore, there is a need to balance the per-module losses within the total loss function. Otherwise, the module with the largest range of values (i.e., the length estimation module) will disproportionately affect the total loss during training, given its larger potential errors. To overcome this challenge, loss weights, denoted as w_{dia} and w_{color}, respectively, were added to the diameter and color modules.

Loss balancing for per-image estimations: The per-image traits of mean diameter and percentage of white roots also have small ranges of values. The percentage of white roots is in the range of 0–1, and 99 % of images have a mean diameter value of less than 1 mm. To promote learning, we examined also adding weights to the modules for the per-image estimating networks. The loss weights for modules (*"D+R"* or *"MSR"*) estimating mean diameter and color (percentage of white roots) are denoted w_{dia} and w_{color}, respectively.

Appendix C1 provides the values chosen for the loss balancing parameters, while Appendix C2 offers additional details on training and evaluation procedures.

2.5. Data

2.5.1. Data acquisition

The data included 663 annotated images of grapevine (Vitis vinifera) roots selected randomly from images acquired on seven days between August 2012 and December 2013. Image acquisition was from 40 tubes across a range of depths in the soil profile at the Ramat Negev Research and Development Center. The studied grapevines consist of two scion cultivars, Shiraz (SH) and Cabernet Sauvignon (CS), grafted onto the same rootstock, Ruggeri 140 (V. berlandieri x V. rupestris). They were examined under different irrigation levels (full and deficient). Observation tubes were either 25 or 75 cm from the trunk, and were 2 m long, with a 5.15 cm inner diameter and 6 cm outer diameter.

Data were acquired with a manual MR acquisition system using an MR camera (BTC-2; Bartz Technology, Carpinteria, California, USA) with its default settings. The camera was attached to an indexing handle and controlled using a laptop-based image capture system (I-CAP; Bartz Technology, Carpentaria, California, USA). Each image captured an area of 16.5 mm × 14 mm (231 mm^2) at a resolution of 754 × 648 pixels.

2.5.2. Image annotation

The annotations, manually obtained with Rootfly and referred to as the ground truth (GT) values, include per-root traits of length, diameter, and color for each root in an image, along with dot annotations (Fig. 2-5a). The dots are point coordinates that the annotator marks along the entire length of each root during the annotation process, allowing Rootfly to estimate root length. These point coordinates are used to generate a GT heat map of points' presence in each location with a Gaussian kernel placed around each annotated point (Fig. 2-5b).

Four categories of root color can be selected in Rootfly: white, light brown, dark brown, and black. To simplify training, the first two categories were annotated as "white", while the latter two were annotated as "dark brown". In addition, using the length, diameter, and color values of each root, we calculated the per-image GT traits, which include TRL, mean diameter, and the percentage of white roots.

2.5.3. Datasets for training and testing

Training and testing used the following subsets as detailed below.

1. ***"all_data"*** includes 531 images from the full grapevine dataset (Table B-1) randomly assigned to training, validation, and test sets containing 383, 46, and 102 images, respectively.
2. ***"limit_5"*** is a variation of the 531 images' annotations that includes annotations only for roots longer than 5 mm; it ignores shorter roots, not annotating them as roots. This dataset was used for alternative training (Section 3.1.2).
3. ***"combined dataset"*** combines into a single dataset the training images from the *"all_data"* dataset with an additional 1712 images from two previously published[2] annotated root datasets (the training and validation sets of *"Dataset 1"* and *"Dataset 2"*). This dataset was used to train the root detection network (Section 3.1.1).

[2] The datasets are publicly available in the Zenodo repository [https://doi.org/10.5281/zenodo.7482146].

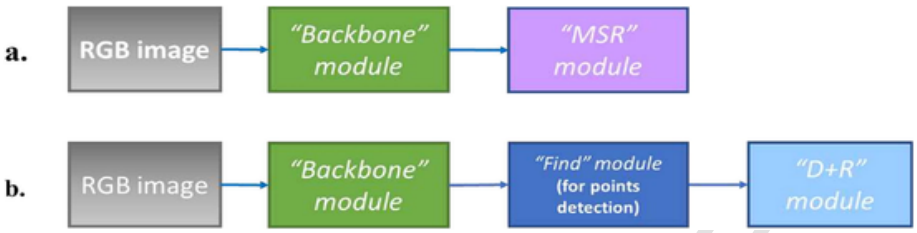

Fig. 2–4. Per-image trait estimation (Khoroshevsky et al., 2024) with. A a regression-based network and b. a points-detection-based network.

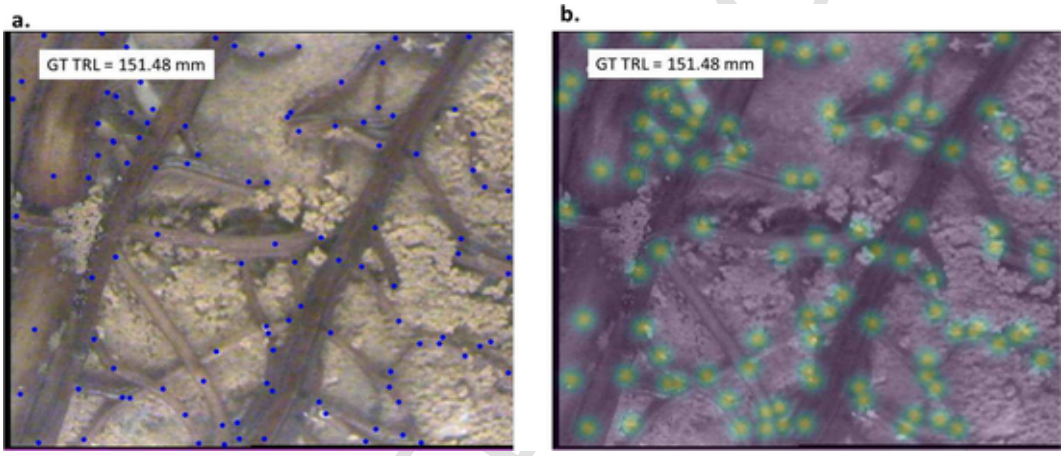

Fig. 2–5. Representative MR image from the grapevine root dataset. **a.** Image with superimposed point annotations (in blue) as obtained from Rootfly. **b.** The generated heat map superimposed on the raw image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. ***"vertical distribution dataset"*** is an additional set of 132 annotated grapevine root images to demonstrate the per-image networks' abilities to evaluate variation in the vertical distribution of roots' characteristics (Section 3.3). It comprises images of different soil depths taken in two specific tubes on a single day in September 2013. The dataset includes 71 images of CS and 61 of SH, grown under full irrigation. Each tube was 75 cm from the trunk.

2.6. Complexity and run time

The networks' computational complexities were calculated using the "thop" library (a dedicated tool to assess complexity values for PyTorch models) (Table C3-1). Results are reported in terms of the number of floating-point operations (FLOPs), measured in billions of FLOPs (GFLOPs), and the number of learnable parameters (params). The table shows that the complexity of each network remains constant across the three hardware configurations used in this study (detailed in Appendix C3). It also lists the average training and testing run times.

2.7. Evaluation metrics

The metrics for evaluating object (root) detection performance were standard precision, recall, and average precision (AP).

Errors in trait estimation were assessed in terms of the mean relative deviation (*MRD*), mean absolute difference (*mean abs_diff*), and fraction of explained variance (Eqs. (1)–(3). In the equations, y_i denotes true observation i, $\hat{y}_i$ the network's estimation for observation i, N the sample size, and $\bar{y}$ the mean of all N (true) observations. The network's estimations for the evaluation set are thus $\{\hat{y}_i\}_{i=1}^{N}$, the set of GT values is $\{y_i\}_{i=1}^{N}$, and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ is the mean GT value. The following outlines the calculation of the relevant error evaluation metric for observation i based on the trait values.

1 Positive observations ($y_i > 0$): This is relevant to per-root estimations of length and diameter and to per-image estimations of TRL and mean root diameter. The MRD metric gives the error (Eq. (1):

$$\text{error} = \text{MRD} = \frac{\sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{y_i}}{N} \quad (1)$$

2 Binary observations: This is relevant to per-root estimation of color (being either white, 1, or dark brown, 0). Eq. (2) gives the classification error:

$$\text{error} = \text{mean abs\_diff} = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \quad (2)$$

3 Non-negative observations in the range of [0,1]: This is relevant to the per-image estimation of the percentage of white roots in an image. As y_i may be 0, the error is measured with the same metric as for the binary case (Eq. (2), using the average absolute difference between the true and estimated values.

4 Fraction of explained variance (Eq. (3): This statistic is used to examine whether the suggested network's performance (its mean squared error is the numerator) is preferable to always estimating the required value using the trivial estimation of the mean $\bar{y}$ (the mean squared error of this estimator is the denominator)

$$1 - \mathrm{FVU} = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \quad (3)$$

2.8. Analysis

Root detection was compared for the two training options: i.e., using the *"all_data"* full grapevine dataset or the *"combined dataset"* (Section 3.1.1). Then the variations of the 'phenotyping' section architectures were examined to analyze the per-root estimations while keeping the root detection section fixed (Section 3.1.2). The selection of the best configuration (network architecture and training data) for each network type (*points-detection-based* or *regression-based*) was performed by training the relevant architectures with either the *"all_data"* or the *"limit_5"* dataset and using the validation set of *"all_data"* for the selection of weights. The per-image traits were evaluated using both our previously published direct estimation networks (Khoroshevsky et al., 2024, Fig. 2-4) and by aggregating the per-root estimations in an image with the newly suggested networks from Fig. 2-2 and Fig. 2-3.

Two additional analyses were performed to find the best networks for per-image trait estimation. The first involved training with different training set sizes while keeping the validation and test sets of *"all_data"* fixed (Section 3.4). The second was to examine the networks' ability to generalize by using *k*-fold cross-validation (with $k = 5$) with the *"all_data"* dataset. Performance was evaluated by calculating the average and standard deviation (std) of the MRD and $1 - FVU$ metrics (Section 3.5).

3. Results and discussion

3.1. Per-root phenotyping

3.1.1. Roots detection

The precision and recall results (~65 % and ~ 47 %, respectively) were similar for training on the "*combined dataset*" and the "*all_data*" datasets, with the AP value being slightly higher (by ~ 0.4 %) for the larger "*combined dataset*". We continued the rest of the evaluations using the latter detector. The recall, precision, and AP results on the test set of "*all_data*" were 67 %, 46 %, and 38 %, respectively. Fig. 3-1 visually exemplifies root detection when the network is trained on the *"combined dataset"*.

3.1.2. Per-root trait estimation

a. Selecting the best configurations

For per-root trait estimation, we evaluated the different training configurations in terms of architecture and training data. The *regression-based* network (Fig. 2-2b) succeeds in learning to estimate all three root traits when the 'phenotyping' section is trained with *"all_data"* (Table D1-1). In contrast, the color module fails to learn correctly in the *points-detection-based* network (Fig. 2-2a): the network is not able to learn the individual roots' colors, and the estimated output is always 1 (white). Once cropped and resized, the images of the small, detected roots input to the 'phenotyping' section of the network have insufficient information for meaningful learning of the heat maps to allow color estimation. Training the *points-detection-based* network (Fig. 2-2 a) with the *"limit_5"* dataset results in successful learning for root color. On the other hand, for both networks, training with the root length limitation (i.e., using the *"limit_5"* dataset) results in the length modules being unable to learn correctly.

Per-root trait estimation by a *points-detection-based* network is improved by using a computationally heavier architecture (Fig. 2-3) composed of a mix of pre-trained modules (Section 2.3.2). Weights for bounding box detection are those obtained when training the detection section on the *"combined dataset"*. The right branch (for diameter and color estimation) consists of modules pre-trained on the *"limit_5"* dataset. The left branch (for length estimation) consists of modules pre-trained on the *"all_data"* dataset.

The best combinations for per-root trait estimation were the *regression-based* network (Fig. 2-2b) trained using the *"all_data"* training set and the *points-detection-based* network composed of a mix of pre-trained modules (Fig. 2-3). Appendix D1 gives detailed results.

b. Per-root trait estimation of the chosen configurations

Comparing the configurations for per-root phenotyping as evaluated on the test set of "*all_data*" (Table 3-1) reveals that both configurations achieve similar error results. Both can infer root length with high precision ($1 - FVU > 0.9$), and provide informative estimates for diameter and color with ~ 15 % *MRD* error values. We also consider the *mean_abs_diff* (measured in mm) to evaluate the diameter estimations (Table 3-1) to emphasize that the seemingly high errors of the diameter estimations, as measured by *MRD*, correspond to very small absolute error values because of the distribution of diameter values (i.e., 99 % of images having a mean diameter smaller than 1 mm).

3.2. Per-image phenotyping and comparison with other works

Table 3-2 presents the results of per-image phenotyping (TRL, mean diameter value, and the percentage of white roots in an image) separately for each trait. These results were obtained either through direct estimation (Khoroshevsky et al., 2024, Fig. 2-4) or by aggregating per-root estimations from the selected per-root configurations (Fig. 2-2b, Fig. 2-3).

Estimation based on per-image networks (Khoroshevsky et al., 2024**,** Fig. 2-4**).** Among the three estimated traits, the best results in terms of low error (13.7 %) and high $1 - FVU$ (0.92) are for TRL estimation with the *points-detection-based* network (Fig. 2-4b), which outperforms the *regression-based* network (Fig. 2-4a) by 2.3 %. The *regression-based* network provides better results than the *points-detection-based* network for estimating the percentage of white roots and mean diameter, with estimation errors lower by 10.3 % and 3.2 %, respectively. The *points-detection-based* network fails to explain a significant fraction of the variance for these tasks, with $1 - FVU$ values of 0.19 and 0.07 for the percentage of white roots and mean diameter estimations, respectively.

Per-root estimation aggregation. For per-image phenotyping, TRL is best estimated with direct per-image phenotyping (Khoroshevsky et al., 2024, Fig. 2-4): using per-root aggregates increased the TRL error by ~ 25 % and ~ 23 % for the *points-detection-based* and *regression-based* networks, respectively. However, for mean diameter and the percentage of white roots, the best results were obtained by aggregating per-root phenotyping estimates using a *regression-based* network (Fig. 2-2b), despite the imperfections of the object detection task. Specifically, aggregating the per-root estimates using a *regression-based* network (Fig. 2-2b) yielded better results than the direct per-image estimates using a *regression-based* network (Fig. 2-4a), with decreases of ~ 4 % and 3.3 % in the estimation error of white roots percentage and mean diameter, respectively. The $1 - FVU$ value for the estimates obtained with the per-root *regression-based* network (Fig. 2-2b) aggregation also shows an improvement of 0.11 relative to per-image estimation (with a *regression-based* network, Fig. 2-4a) for mean diameter estimation; it remains the same (0.69) for the percentage of white roots.

Following these results (Table 3-2), we recommend using our previous networks (Khoroshevsky et al., 2024, Fig. 2-4) for TRL estimation. Mean diameter and percentage of white roots are best estimated by aggregating per-root trait estimations, as they are less affected by imper-

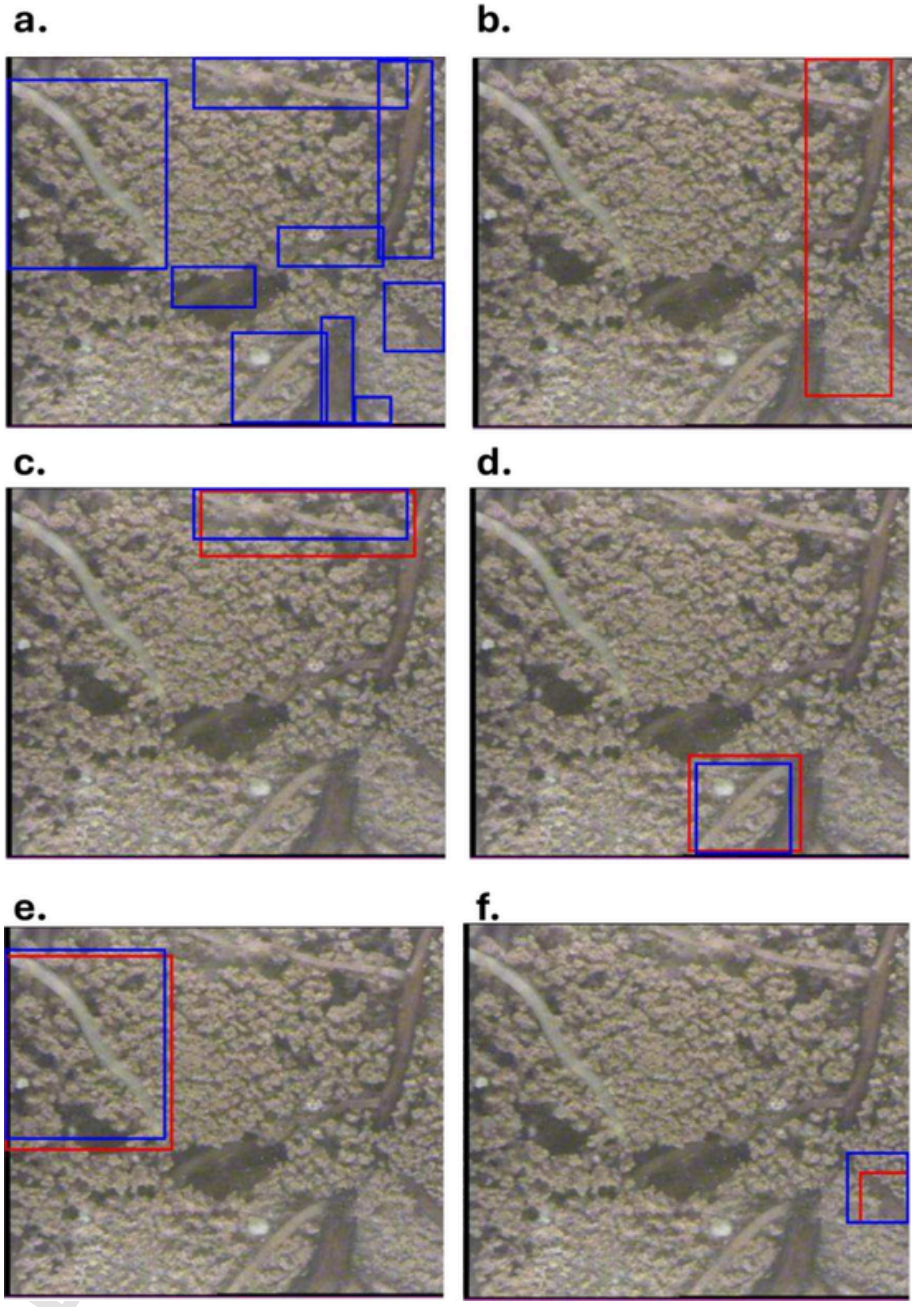

Fig. 3–1. Representative grapevine root detection from the "all_data" validation set; training used the "combined dataset" training set, including **a.** all the GT bounding boxes in blue, **b.** false positive detection in red, and **c.–f.** true positive detection in red with the corresponding GT bounding box in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fect root detection, whereas estimating TRL based on both true and false root detections led to poor estimation results.

In addition, we compare our best results for TRL estimation with those obtained using segmentation-based tools (Bauer et al., 2022; Smith et al., 2022; Baykalov et al., 2023), based on their own datasets (detailed in Appendix D2). Following those previous works, we use the coefficient of determination R^2 to assess TRL estimation. Our suggested approach achieves a high R^2 value of 0.93, which is on par with the results previously reported for TRL estimation (Table D2-1). However, our main contribution is the added ability to estimate root diameter and color, which has not been reported before.

3.3. Vertical distribution of root traits

Per-image traits were estimated for an additional set of images of grapevine roots (scion varieties SH and CS); this *"vertical distribution dataset"* (Fig. 3-2) comprised data from two different tubes obtained at the end of the experiment. The values of per-image traits—root length density (RLD), mean root diameter, and white root percentage (based on roots' length, diameter, and color traits)—at different soil depths (Fig. 3-2) can indicate the vertical root distribution, with RLD being calculated from the TRL of a specific imaging area (in cm/cm^2).

These estimation curves (Fig. 3-2) are generated based on the recommended trained networks of Section 3.2 (specifically the trained networks in Fig. 2-4b for TRL estimation, and aggregating the per-root esti-

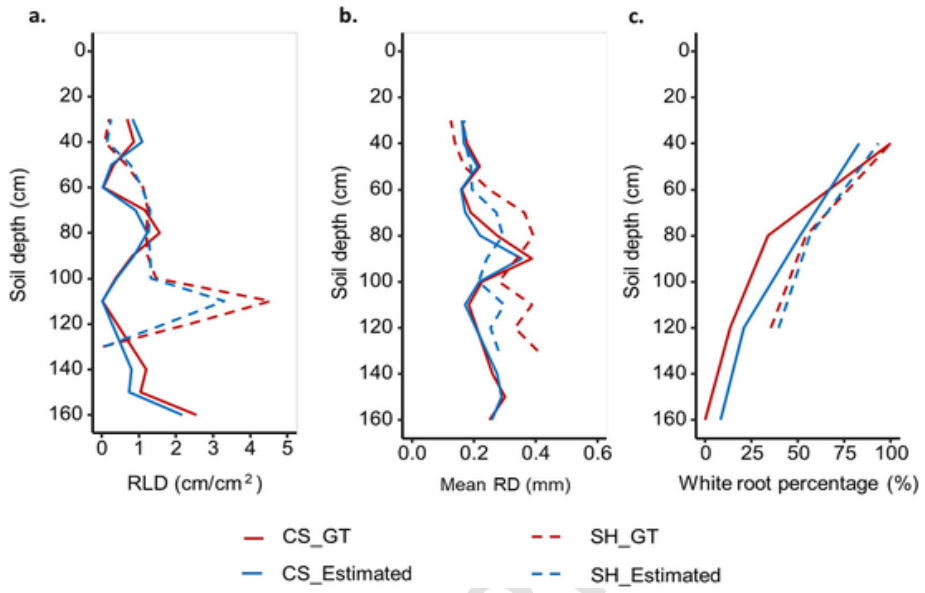

Fig. 3–2. Comparison of root trait estimation using the recommended networks for per-image trait estimation vs. the GT values for root images of two different scion varieties (CS and SH) when estimating **a.** RLD, **b.** mean root diameter, and **c.** percentage of white roots varying with soil depth.

Table 3–1
Summary of errors for per-root trait estimation on the test set of "all_data" using the best configurations of per-root phenotyping with regression-based and points-detection-based networks.

	'phenotyping' section	**Color (binary)**		**Length (mm)**		**Diameter (mm)**		
	Training data	Error % (*mean abs_diff*)	*1-FVU*	Error % (*MRD*)	*1-FVU*	Error % (*MRD*)	*1-FVU*	Error mm (*mean abs_diff*)
Fig. 2-2b	*"all_data"*	8.8	0.63	15.5	0.91	23.5	0.60	0.086
Fig. 2-3	Mix of pre-trained modules	9.1	0.62	14.9	0.92	25.0	0.52	0.095

mates of Fig. 2-2b for mean root diameter and white root percentage estimation). A qualitative comparison of these curves to those derived from GT annotations (Fig. 3-2) indicates the alignment between the GT and the estimated root trait values.

3.4. Different sample sizes

Increasing the number of training images improves each network's ability to estimate traits (Table 3-3). This improving trend was maintained across the tested range of training set sizes.

3.5. Evaluation by k-fold cross-validation

Table 3-4 provides the networks' average error estimates based on 5-fold cross-validation. The TRL, percentage of white roots, and mean root diameter remain stable across the different data splits, with std values of 1.7, 2.0, and 2.2, respectively.

The resulting error (MRD) and 1 – *FVU* values are similar to those reported for the original test set (Table 3-2), with the average error from 5-fold cross-validation being ~ 2 % higher than that of the original test set for each of the estimated traits.

4. Conclusions

This study presents a new framework based on CNNs for the automatic, non-destructive, and reliable estimation of root traits from *in situ* images. The process does not require the training of a segmentation network as a preliminary step to trait estimation. The annotated dataset of MR images is made publicly available as part of this paper.

The network architectures proposed in this framework were developed using a suggested set of modules to estimate per-root traits (color, length, and diameter) and per-image traits (TRL, mean root diameter, and percentage of white roots).

A particular innovation of this study is the automated root color estimation. Additionally, while segmentation-based tools proposed in the literature claim the ability to estimate root diameter, such results have not been reported. In this study, the diameter estimates from our net-

Table 3–2
Results on the "all_data" test set for each per-image trait and each estimation network are presented. Values in bold are the errors for each trait based on the better estimation approach: i.e., using per-image networks vs. per-root estimation aggregation.

		TRL (mm)		**% of white roots**		**mean root diameter (mm)**		
		Error % (*MRD*)	*1-FVU*	Error % (*mean abs_diff*)	*1-FVU*	Error % (*MRD*)	*1-FVU*	*Error mm (mean abs_diff)*
Per-image direct estimation	Fig. 2-4a (Khoroshevsky et al., 2024)	16.0	0.92	15.3	0.69	20.9	0.31	0.082
	Fig. 2-4b (Khoroshevsky et al., 2024)	**13.7**	**0.92**	25.6	0.19	24.1	0.07	0.093
Per-root estimation aggregation	Fig. 2-2b	38.9	0.40	**11.5**	**0.69**	**17.6**	**0.42**	0.070
	Fig. 2-3	38.6	0.42	16.5	0.46	22.1	0.12	0.090

Table 3–3
Results for the test set "all_data" after using different dataset sizes to train the recommended networks for per-image root phenotyping (specifically the networks of Fig. 2-4b for direct TRL estimation, and aggregating the per-root estimates of Fig. 2-2b for mean root diameter and white root percentage estimation).

Training set size	**TRL (mm)** (*points-detection-based* direct estimation, Fig. 2-4b)		**% of white roots** (*regression-based* per-root aggregation, Fig. 2-2b)		**mean root diameter (mm)** (*regression-based* per-root aggregation, Fig. 2-2b)		
	Error % (*MRD*)	*1-FVU*	Error % (*mean abs_diff*)	*1-FVU*	Error % (*MRD*)	*1-FVU*	Error mm (*mean abs_diff*)
100 images	20.3	0.87	22.5	0.26	23.2	0.01	0.096
200 images	16.1	0.92	18.0	0.44	20.9	0.16	0.086
300 images	15.0	0.94	15.4	0.51	20.2	0.19	0.085
Full set (383 images)	***13.7***	**0.92**	**11.5**	**0.69**	**17.6**	**0.42**	**0.070**

Table 3–4
Summary of 5-fold test results for the recommended networks used in per-image root phenotyping.

	Error (MRD) % *(Average, std)*	*1-FVU (Average, std)*
TRL (mm) (*points-detection-based*, Fig. 2-4b)	15.6, 1.7	0.94, 0.02
% of white roots (*regression-based* per-root aggregation, Fig. 2-2b)	13.7, 2.0	0.59, 0.08
mean root diameter (mm) (*regression-based* per-root aggregation, Fig. 2-2b)	19.8, 2.2	0.34, 0.16

works were compared with manual measurements derived from root annotations, alongside length and color traits.

The proposed networks in this framework are either *regression-based* or *points-detection-based*. The *regression-based* networks need only the value of the relevant trait for training. The *points-detection-based* networks are based on the detection of points marked along each root by the user during the annotation process in Rootfly. These networks enable the visualization of root locations via the output of heat maps of detected points, and thus can assist the annotation of new root images.

A demonstration of our proposed system involved the qualitative assessment of vertical distributions of roots' length, diameter, and color in two different grapevine scion varieties. This could aid researchers in selecting optimal grafting combinations for specific environments.

Our proposed framework could be useful for training networks to estimate other root traits (or traits of any other objects) in scenarios involving per-object and/or per-image tasks. It benefits from CNN-based methods without requiring pre-segmentation.

Limitations of the current work relate to the method's ability to be used generally with new datasets. Performance may be reduced when using other camera types and different root and soil colors. To address this, the networks can be further trained with additional datasets acquired from diverse conditions to improve their generalized applicability.

Uncited references

.

CRediT authorship contribution statement

Faina Khoroshevsky: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Kaining Zhou:** Writing – review & editing, Visualization, Data curation, Conceptualization. **Aharon Bar-Hillel:** Writing – review & editing, Methodology, Conceptualization. **Ofer Hadar:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Shimon Rachmilevitch:** Data curation, Conceptualization. **Jhonathan E. Ephrath:** Writing – review & editing, Supervision, Conceptualization. **Naftali Lazarovitch:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Yael Edan:** Writing – review & editing, Supervision, Methodology, Conceptualization.

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.

Acknowledgments

This research was funded by the Israeli Ministry of Agriculture and Food Security (grant number 16-38-0044). Partial support was provided by the W. Gunther Plaut Chair in Manufacturing Engineering at Ben-Gurion University of the Negev, Israel.

Data availability

The dataset is publicly available in the Zenodo repository [https://zenodo.org/record/8084106]

Appendix A. Module architecture

1.1. "Backbone" module

A *"Backbone"* module can have many off-the-shelf and pre-trained weight options. The aim of our *"Backbone"* module is to generate feature-rich representations of an image for use as inputs for the subsequent modules. Here, it is based on applying ResNet-50 (He et al., 2016) as a dense convolutional network, and then applying a FPN (Lin et al., 2017a) on top of it, which generates five representation tensors for the original image (Fig. A1-1). The ResNet-50 network of the *"Backbone"* modules is used with initial weights (pre-trained on the ImageNet; Deng et al., 2009) and fine-tuned during the training of the relevant network in which it was used. Its three output maps (C3, C4, and C5) are used as the inputs for the FPN module. The FPN module generates 5-scale feature pyramid representations (designated as output tensors $P_3 - P_7$) of the original input image on multiple scales. These tensors include five representations of the original image in multiple octaves, with the spatial dimensions of P_i being half those of P_{i-1}. Each representation has 256 maps, having a spatial size of $\frac{s}{2^i}$, where s is the input image size. We use either all five image representations ($P_3 - P_7$) as inputs for the subsequent modules or only P_3 (that with the largest resolution), depending on the specific network being composed. As detailed

below, when the outputs of the *"Backbone"* module are used as inputs to the *"MSR"* module or as inputs to the *"Find"* module as part of an object detection network, all five representations are relevant. When the outputs of the *"Backbone"* module are used as inputs to the *"Find"* module in single estimation networks (for counting, length, diameter, and color estimations using the *"D+R"* module), we use P_3 alone.

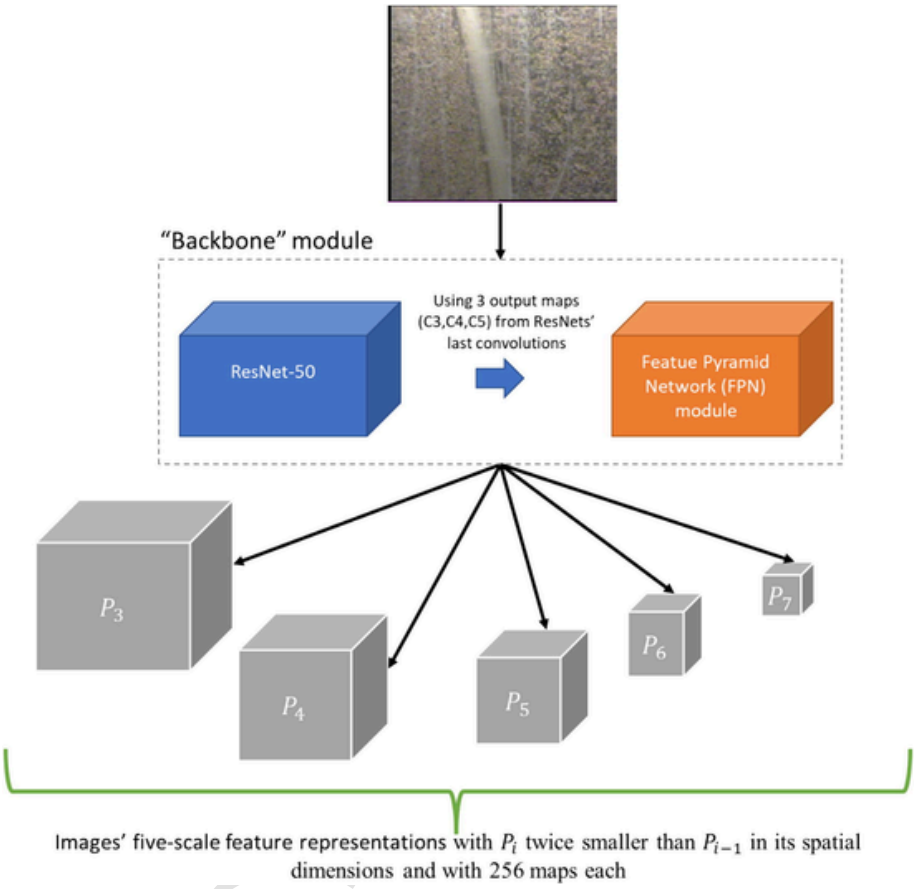

Fig. A11. Illustration of the "Backbone" module.

1.2. *"Find" module*

This module always outputs heat maps (one or several), which spatially locates points' presence in an input tensor. The maps represent the probability of an object's presence at each spatial location with a value between 0 and 1. The outputs are of the same resolution as the input tensor. The identified points can represent either the center of an object, like in a counting task (Khoroshevsky et al., 2021), or annotations along the roots in an image, like in the root-related tasks. When this module is used as part of an object detection network, it generates nine heat maps for each position of the input tensor, and the probability of an object's presence is estimated for nine predefined anchor rectangles. When used for detecting points' presence, it outputs a single heat map of the probability of a point's presence at each location.

Output modes. The *"Find"* module can be used in two output modes depending on its purpose in the composed networks.

- ***"Find (for bbox detection)"*** implements the fully convolutional classification subnetwork of the RetinaNet architecture (Lin et al., 2017b) to estimate for each class the probability of an object's presence at each spatial position of the input tensor (Fig. A21). The inputs for this module are the five pyramid tensors produced by the FPN of the *"Backbone"* module. It processes each of them independently using the *"FeatureClassification"* sub-module (detailed below), and thus produces five output tensors. Each output is a tensor with the same spatial dimensions as the input tensor, but with nine probability maps.
- ***"Find (for points detection)"*** is used for *points-detection-based* networks. It outputs a single heat map with a general estimation of the probability points' locations (Fig. A22). In this mode, the module has two sub-variations depending on its outputs for the subsequent *"D+R"* module, when attribute estimation is required. These are the *"Find (for counting)"* variation, which is specifically relevant to an object's part counting task (Khoroshevsky et al., 2021), and the *"Find (for any attribute)"* variation, which is relevant to the other estimations.

This mode operates only on the high-resolution pyramid scale (P_3). Like in the *"Find (for detection)"* variation, an input tensor initially goes through the *"FeatureClassification"* module, but unlike the *"Find (for detection)"* option, for a given class, this variation of the module generates a single output map (not nine) of the same resolution as the input tensor. The map states the probability of a point's presence at each location. This mode of the *"Find"* module is intended to be relevant to estimating some attribute (e.g., object count, length, diameter, and color) that are based on the detecting points. In such cases, this module is part of a larger network, preceding a *"D+R"* module (detailed below). It is specifically relevant to two sub-cases of attribute estimation: count estimation, *"Find (for counting)"*, and any other attribute estimation, *"Find (for any attribute)"*. For

the case studies in this paper, the *"any attribute"* sub-case is demonstrated in the per-root phenotyping task, in which the attributes are length, diameter, and color.

***"FeatureClassification"* sub-module.** In both output modes, this sub-module generates heat maps. To learn to generate the output maps, there is for each generated map a ground truth heat map that is created in training, with a Gaussian kernel placed around each annotated point. It is trained with Focal Loss (Lin et al., 2017b), which is designed to address the imbalance between foreground and background classes during training. This sub-module outputs a tensor F with the same spatial dimensions as its input tensor (one of the $P_3 - P_7$ tensors). The depth of F depends on the specific task for which this module is used. The module consists of four convolutional layers, each with 256 filters of size 3×3 and a ReLU activation, culminating in the output tensor F with a fifth 3×3 convolutional layer. The number of filters and the activation function of the fifth layer depend on the specific task for which this module is used:

- ***"Find (for bbox detection)"*:** the fifth convolutional layer has nine filters, and its activation function is sigmoidal.
- ***"Find (for points detection)"*:** the fifth layer has a single filter and a ReLU activation function, outputting a single probability map. In addition, following the work of Itzhaky et al. (2018), there are also guiding internal layers that are created by estimating an intermediate heat map after each of the first four convolutional layers. Each intermediate layer is an additional convolutional layer and a ReLU activation with a single 1×1 filter. The intermediate guiding heat maps are then created with decreasing kernel size, such that the heat map regression task is cruder and simpler in the initial layers, forcing each generated heat map to mimic the ground truth heat map with additional regression losses.

Additional layers. The *"Find"* module has additional layers that are relevant only to the *"Find (for points detection)"* mode. In addition to these heat maps, the module also outputs the last 256-depth feature tensor from the *"FeatureClassification"* module (the input for the fifth convolutional layer). The fifth output heat map of the *"FeatureClassification"* module is subjected to smooth non-maximum suppression, keeping the activity of the values close to 1, while all other values in the map are reduced to 0. We refer to this map as map M. The coordinates of the non-zero values in map M are the estimated points' coordinates.

The next layers depend on the application: i.e., whether the module is used for counting (Khoroshevsky et al., 2021) or a different estimation.

- ***"Find (for counting)"***: A global sum operation is then applied on top of map M, giving an initial detection-based estimation of the count value.
- ***"Find (for any attribute)"***: The *"Find"* module is adapted to be a part of a network that eventually outputs an estimation that is not necessarily a count value and can be relevant to any other traits such as a binary value for root color. Map M is input to an additional 3×3 convolutional layer with 256 filters and a ReLU activation, followed by a 3×3 convolutional layer with a single filter and a sigmoid activation. This map is flattened to a vector and uses a linear layer and a sigmoid activation to output a single value. This value is an initial detection-based estimation when the module is used to generate a binary output (e.g., color in our case), but when it is used for other non-binary estimations (e.g., per-root length or diameter), it is only an additional feature that will be used for the final estimation of the relevant trait in the subsequent *"D+R"* module together with the 256-depth feature tensor from the *"FeatureClassification"* module.

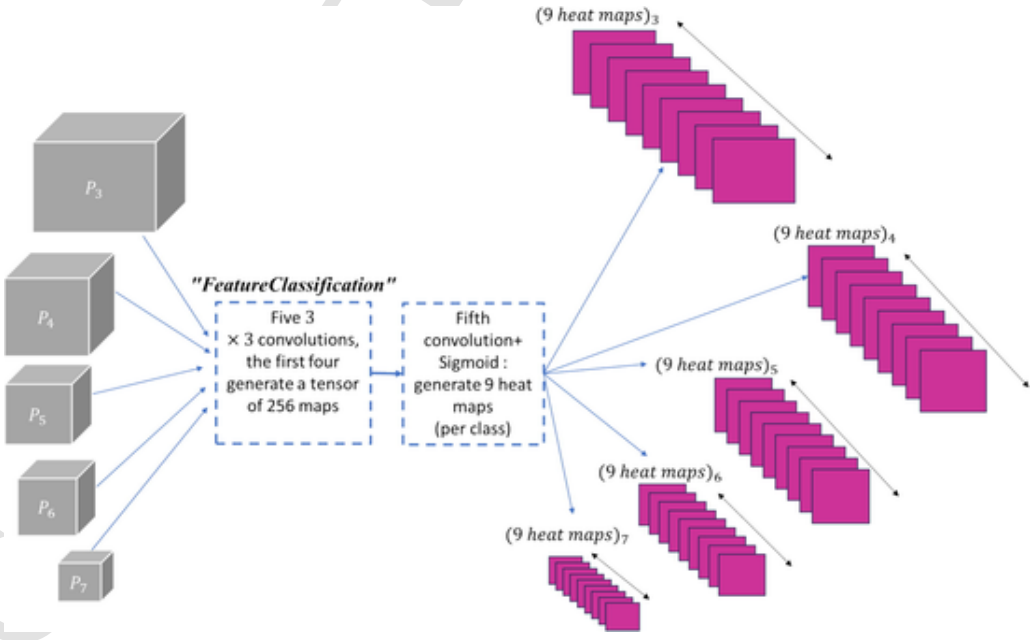

Fig. A21. Illustration of the "Find" module in the "Find (for bbox detection)" mode.

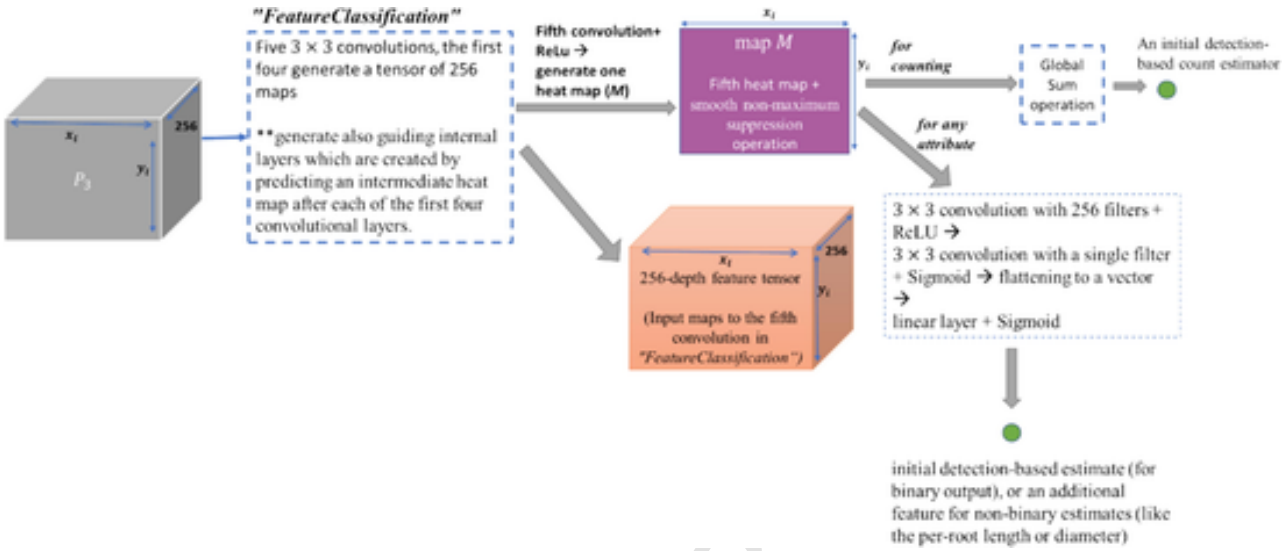

Fig. A22. Illustration of the "Find" module in the "Find (for points detection)" mode.

1.3. *"Where" module*

This module is relevant to object detection (e.g., for parts-per-object counting or per-root phenotyping) in the composed network. Similar to the *"Find (for detection)"* module, this module processes all the pyramid tensors $P_3 - P_7$ and produces five output tensors by implementing the bounding box regression sub-network of RetinaNet. For each input tensor, it estimates for every position and possible anchor (among the nine considered), a four-dimensional bounding box refinement vector. The vector includes the corrections required for the bounding box to improve its match to the object in terms of its parameters (x, y, width, and height). Inference is achieved by applying, on each of the inputs, five convolutional layers, where each keeps the same spatial dimensions as the input pyramid tensor. The first four layers have 256 filters each, and the fifth layer has four filters (so the output includes 36 maps overall). It is trained by propagating a smooth-L1-regression loss matching the estimated values to ground-truth rectangles—but only for anchors with relevant objects.

1.4. *"RoI-Align" module*

This module is relevant when both object detection and per-object estimations are required, because it crops and resizes the detected objects from the original image for further use (He et al., 2017). The 'crops' are passed to the next section of the relevant network, in which each 'crop' is treated as an independent input image. Each 'crop' is assumed to contain a single detected object. When used for parts-per-object counting (i.e., the number of parts per object), the next section outputs the parts count for that object, and when used for the per-root traits the next section outputs the length, diameter, and color for that object. The *"RoI-Align"* module receives as input the original image and a set of rectangle coordinates of n_d detected objects. By re-sampling the image according to the given coordinates, it outputs a tensor of size $s \times s \times n_d$ of image 'crops', where the new object size (empirically set to 640) is s. Such resizing facilitates the handling of significant variation in sizes of objects of the different datasets without the need for a data-specific configuration.

1.5. *"MSR" module*

This module is based on multiple-scale regression (designated ***"MSR"***) used for *regression-based* estimations for the different phenotyping tasks, for parts-per-object counting, for per-image estimations, and for per-root phenotyping. It has two modes: *"MSR (for binary)"* for a binary output (e.g., root color) and *"MSR (for continuous)"* for continuous outputs (e.g., counting and estimating length and diameter). It can receive all, or a subset of, the representation tensors $P_3 - P_7$ generated by the *"Backbone"* module, such that multiple estimations are produced based on different resolutions; the one with the lowest estimated variance for the input image is chosen. This regression module includes two 3×3 convolutional layers and a ReLU activation, with 256 output maps, where each keeps the spatial dimensions of its input, followed by global average pooling (GAP), flattening the maps to a 256×1 representation vector. This 256-feature vector is fed into two fully connected layers of decreasing size, 128 and 64. In *"MSR (for binary)"* mode, each convolution is followed by a sigmoid activation, and in "*MSR (for continuous)*" mode, it is followed by a ReLU activation. Using an additional fully connected layer, this module outputs two neurons. The first estimates the expected trait value (e.g., part count or root length) and in "*MSR (for binary)*" mode a sigmoid is applied to it. The second estimates the variance of the error expected in the estimation of the phenotype value, using the loss function suggested by Kendall and Gal (2017). Among the estimations based on the considered resolutions of the input representation tensors, the estimation with the lowest estimated variance for the input image is chosen.

1.6. *"D + R" module*

The *detection and regression-based* module (designated ***"D + R"***) is used in two output modes: *"D + R (for binary)"* and *"D + R (for continuous)"* as detailed below. The *"D + R (for continuous)"* option is relevant to any continuous output such as part counting, TRL estimation, and per-root length or diameter. The *"D + R (for binary)"* is relevant to any binary output such as root color. In all cases, it is trained with an L1-regression loss.

The *"D+R"* module gets from the *"Find"* module the following: 1) an initial detection-based estimation of the attribute value (in the *"for counting"* variation) or an additional feature (in the *"for any attribute*" variation), and 2) the feature tensor F_1 preceding it (the 256-depth feature tensor from the *"FeatureClassification"* module outputs).

The final relevant estimation is based on all these inputs, with a distinction being made between a binary output (e.g., color) and a continuous output (e.g., count, length, or diameter). Only when a binary output is required does the tensor F_1 go through additional 3×3 convolutional layers with 256 filters and a sigmoid activation. This additional convolution allows the feature values to be negative, as the F_1 tensor was generated with a ReLU activation. A GAP layer is subsequently applied to extract a 256-feature vector, which is then concatenated with the initial detection-based estimation/additional feature. The final phenotype estimation is computed by linear regression from this 257-feature vector: for binary output, the linear layer is followed by a sigmoid activation, and for continuous output, the linear layer is followed by a ReLU activation.

Appendix B. dataset details

XXX

Table B-1
The full grapevine roots dataset ("all_data") information.

Data type	Set	Number of images	Number of roots	Roots per color (% of total roots)
All roots	Train	383(36 images without roots)	2,920	White 1212 (41.5 %),Dark brown 1708 (58.5 %)
	Validation	46(6 images without roots)	313	White 127 (40 %),Dark brown 186 (60 %)
	Test	102(8 images without roots)	878	White 305 (35 %),Dark brown 573 (65 %)
	Total	531 (481 images with roots,50 without roots)	4,111	

Appendix C. additional training information

3.1. Loss balancing

Both in the per-root and per-image phenotype estimations, the values of w_{dia} and w_{color} were empirically chosen based on the minimal estimation errors obtained on the validation set. The tested values for w_{dia} and w_{color} were 10, 100, and 1000. For per-root estimations, we trained both the *regression-based* modules (*"MSR"*) and *points-detection-based* modules (*"D+R"*) with loss weights of $w_{dia} = 10$ and $w_{color} = 100$. For per-image estimations, the *points-detection-based* networks were trained with $w_{color} = 100$ and $w_{dia} = 1000$ for the percentage of white roots and mean diameter estimation, respectively. For the *regression-based* modules of these tasks, the values were set as $w_{color} = 10$ and $w_{dia} = 10$.

3.2. Training and evaluation

- All networks were trained for 300 epochs. The best epoch was chosen as that with the lowest average error as measured on the validation set of *"all_data"*. In the per-root phenotyping tasks, it was chosen based on the lowest length estimation error. Final performance evaluation was done on the test set of *"all_data"*.
- Root bounding boxes were detected with the aspect ratios parameters of the anchors (the predefined bounding boxes of different shapes and sizes used in RetinaNet) set to [0.5, 1, 3]. The detection performance was examined with an intersection over union threshold of 0.5 and a confidence score threshold of 0.7.
- For the per-image phenotype estimation (TRL, RLD, mean diameter, and percentage of white roots), only images with roots and for which the models yielded root detections were considered. This was done to evaluate the MRD values and to compare the results of direct per-image phenotype estimation with the method of aggregating per-root estimates.
- Whereas per-root estimation is evaluated only on successful root detections (true positives), the per-image estimation variations based on aggregating per-root estimations consider both true and false root detections to provide a realistic method for estimation.

3.3. Complexity, run time, and hardware components

Network examinations were conducted using three machines with NVIDIA GeForce GPUs with the following configurations: a.) RTX 2080 Ti GPU, AMD Ryzen Threadripper 2920X CPU, CUDA 11.3, and PyTorch 1.2; b.) GTX 1080 Ti GPU, Intel(R) Core(TM) i7-7700 CPU, CUDA 12.1, and PyTorch 2.3.1; and c.) RTX 3090 Ti GPU, Intel(R) Core(TM) i9-10900 CPU, CUDA 12.1, and PyTorch 2.3.1. We used a batch size of 1 both for training and inference.

As not all experiments were conducted on all machines, we report the average training and testing run times for the machines the experiments were performed on.

Table C3-1
Complexity in terms of FLOPs, number of trainable parameters, and run time of each suggested network.

Task type	*Suggested network*	*FLOPs* (GFLOPs)	*Params (Million)*	*Training Run Time* (sec\image)	*Testing run time* (sec\image)
Per-Root	Fig. 2-2a (*points-detection-based*)	219.017	70.801	1.4	1.1
	Fig. 2-2b (*regression-based*)	230.264	71.500	1.5	1.3
	Fig. 2-3 (*points-detection-based*)	274.806	104.681	2.5*	1.4
Per-image	Fig. 2-4a (*regression-based*)	98.872	32.727	0.5	0.82
	Fig. 2-4b (*points-detection-based*)	108.738	33.869	0.5	0.94

* Sum of training times of the network on the *"all_data"* and *"limit_5"* datasets.

Appendix D. additional results

4.1. Per-root phenotyping

The *regression-based* network succeeds in learning to estimate all three phenotypes when the 'phenotyping' section is trained with *"all_data"* (Table D1-1). This is in contrast to the *points-detection-based* network, whose color module failed in learning with an estimation error that was ~ 42 % higher than that for the *regression-based* network (Table D1-1). We handled this problem by creating the *"limit_5"* dataset, which included only roots annotations above a certain length. On the other hand, training the *regression-based* network with the *"limit_5"* data resulted in a ~ 5 % increase in the color estimation error as compared with training it with the "*all_data*" training set (Table D1-1). For both networks, training with the root-length limitation (i.e., the *"limit_5"* dataset) resulted in the length modules being unable to learn (error values of 61.4 % and 55.7 % for *points-detection-based* and *regression-based* networks, respectively; Table D1-1). Both network architectures cannot correctly estimate the length of small roots (< 5 mm) if trained on the dataset containing only longer roots (Table D1-2). Lower length estimation errors are obtained when each network is trained on the *"all_data"* dataset, with decreases of ~ 48 % and ~ 43 % in the estimation errors of the *points-detection-based* and *regression-based* networks, respectively. This pre-trained network (Fig. 2-3) improved the estimation results of the *points-detection-based* network (trained on *"limit_5"*) for both length and color, resulting in error reductions of 47.5 %, 2.6 %, and 5.7 %, for the estimations of per-root length, color, and diameter, respectively (Table D1-1).

Table D1-1
Per-root phenotype estimation results on the validation set of "all_data". Bold type denotes the best results for each architecture type, points-detection-based and regression-based.

	Training dataset	Color error % (*mean abs_diff*)	Length error % (*MRD*)	Diameter error % (*MRD*)
points-detection-based network (Fig. 2-2a)	*"all_data"*	53.0	12.9	35.0
	"limit_5"	12.8	61.4	32.4
regression-based network (Fig. 2-2b)	***"all_data"***	**11.4**	**13.0**	**25.7**
	"limit_5"	16.1	55.7	27.1
New composition *points-detection-based* (Fig. 2-3)	**Mix of pre-trained modules**	**10.2**	**13.9**	**26.7**

Table D1-2
Details of length estimation MRD errors by true root length.

		***points-detection-based* network**		***regression-based* network**	
number of detected roots	**GT length range of the detected roots**	**trained on all data (%)**	**trained on roots > 5 mm (%)**	**trained on all data (%)**	**trained on roots > 5 mm (%)**
5	< = 1 mm	9.5	569.3	10.3	494.8
9	[1,2] mm	18.8	270.2	13.7	264.5
12	[2,3] mm	17.4	109.5	15.9	100.2
25	[3,4] mm	17.5	57.1	17.1	48.9
8	[4,5] mm	12.3	24.97	10.2	17.7
90	> 5 mm	10.7	10.3	11.7	9.8

4.2. TRL estimation comparison with segmentation-based works

XXX

Table D2-1
Comparison of TRL estimation results for the test set "all_data" with results reported in previous works based on their own datasets.

Method	Dataset	R^2
Current study –Fig. 2-4a. and Fig. 2-4b	531 images of grapevine roots	**0.93 (for both networks)**

Method	Dataset	R^2
Khoroshevsky et al. (2024) (using the same networks as in Fig. 2-4)	4015 images of the roots of four crop species (corn, pepper, melon, and tomato)	0.88–0.99/0.84–0.96 (same/unseen data type)
Baykalov et al. (2023)	2388 root images of the roots of seven crop species (corn, tomato, grapevine, olive, tree–grass, soybean, and chicory)	0.95–0.96/0.54–0.86(same/unseen data type)
Smith et al. (2022)	857 images of chicory roots	0.89–0.92
Bauer et al. (2022)	36,500 images of wheat (*Triticum aestivum)* and corn (*Zea mays*)	0.50–0.81

References

Aidoo, M.K., Sherman, T., Ephrath, J.E., Fait, A., Rachmilevitch, S., Lazarovitch, N., 2018. Grafting as a method to increase the tolerance response of bell pepper to extreme temperatures. Vadose Zone J. 17 (1), 1–8. https://doi.org/10.2136/vzj2017.01.0006.

Bauer, F.M., Lärm, L., Morandage, S., Lobet, G., Vanderborght, J., Vereecken, H., Schnepf, A., 2022. Development and validation of a deep learning based automated minirhizotron image analysis pipeline. Plant Phenomics. https://doi.org/10.34133/2022/9758532.

Baykalov, P., Bussmann, B., Nair, R., Smith, A.G., Bodner, G., Hadar, O., Lazarovitch, N., Rewald, B., 2023. Semantic segmentation of plant roots from RGB (mini-) rhizotron images—generalization potential and false positives of established methods and advanced deep-learning models. Plant Methods 19 (1), 122. https://doi.org/10.1186/s13007-023-01101-2.

Comas, L.H., Eissenstat, D.M., Lakso, A.N., 2000. Assessing root death and root system dynamics in a study of grape canopy pruning. New Phytol. 147 (1), 171–178. https://doi.org/10.1046/j.1469-8137.2000.00679.x.

Cui, Z., Wu, G.L., Huang, Z., Liu, Y., 2019. Fine roots determine soil infiltration potential than soil water content in semi-arid grassland soils. J. Hydrol. 578, 124023. https://doi.org/10.1016/j.jhydrol.2019.124023.

Danilevicz, M.F., Bayer, P.E., Nestor, B.J., Bennamoun, M., Edwards, D., 2021. Resources for image-based high-throughput phenotyping in crops and data sharing challenges. Plant Physiol. 187 (2), 699–715. https://doi.org/10.1093/plphys/kiab301.

Dannoura, M., Kominami, Y., Oguma, H., Kanazawa, Y., 2008. The development of an optical scanner method for observation of plant root dynamics. Plant Root 2, 14–18. https://doi.org/10.3117/plantroot.2.14.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255. 10.1109/CVPR.2009.5206848.

Farjon, G., Huijun, L., Edan, Y., 2023. Deep-learning-based counting methods, datasets, and applications in agriculture: a review. Precis. Agric. 24, 1683–1711. https://doi.org/10.1007/s11119-023-10034-8.

Galdos, M.V., Brown, E., Rosolem, C.A., Pires, L.F., Hallett, P.D., Mooney, S.J., 2020. Brachiaria species influence nitrate transport in soil by modifying soil structure with their root system. Sci. Rep. 10 (1), 5072. https://doi.org/10.1038/s41598-020-61986-0.

Geng, L., Li, L., Sheng, W., Sun, Q., Yang, J., Huang, Q., Lv, P., 2023. Compound minirhizotron device for root phenotype and water content near root zone. Comput. Electron. Agric. 205, 107592. https://doi.org/10.1016/j.compag.2022.107592.

Gillert, A., Peters, B., von Lukas, U.F., Kreyling, J., 2021. Identification and measurement of individual roots in minirhizotron images of dense root systems. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1323–1331. 10.1109/ICCVW54120.2021.00153.

Han, E., Kautz, T., Perkons, U., Uteau, D., Peth, S., Huang, N., Horn, R., Köpke, U., 2015. Root growth dynamics inside and outside of soil biopores as affected by crop sequence determined with the profile wall method. Biol. Fertil. Soils 51, 847–856. https://doi.org/10.1007/s00374-015-1032-1.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778. 10.1109/CVPR.2016.90.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969. 10.1109/ICCV.2017.322.

Hendrick, R.L., Pregitzer, K.S., 1992. Spatial variation in tree root distribution and growth associated with minirhizotrons. Plant and Soil 143, 283–288. https://doi.org/10.1007/BF00007884.

Huang, Y., Yan, J., Zhang, Y., Ye, W., Zhang, C., Gao, P., Lv, X., 2023. Automatic segmentation of cotton roots in high-resolution minirhizotron images based on improved OCRNet. Front. Plant Sci. 14, 1147034. https://doi.org/10.3389/fpls.2023.1147034.

Itzhaky, Y., Farjon, G., Khoroshevsky, F., Shpigler, A. and Bar-Hillel, A., 2018, September. Leaf counting: Multiple scale regression and detection using deep CNNs. In BMVC (Vol. 328).

Iversen, C.M., McCormack, M.L., Powell, A.S., Blackwood, C.B., Freschet, G.T., Kattge, J., Roumet, C., Stover, D.B., Soudzilovskaia, N.A., Valverde-Barrantes, O.J., van Bodegom, P.M., 2017. A global Fine-Root Ecology Database to address below-ground challenges in plant ecology. New Phytol. 215 (1), 15–26. https://doi.org/10.1111/nph.14486.

Johnson, M.G., Tingey, D.T., Phillips, D.L., Storm, M.J., 2001. Advancing fine root research with minirhizotrons. Environ. Exp. Bot. 45 (3), 263–289. https://doi.org/10.1016/S0098-8472(01)00077-6.

Kendall, A., Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision? Adv. Neural Inf. Proces. Syst. 30.

Khoroshevsky, F., Khoroshevsky, S., Bar-Hillel, A., 2021. Parts-per-object count in agricultural images: Solving phenotyping problems via a single deep neural network. Remote Sens. (basel) 13 (13), 2496. https://doi.org/10.3390/rs13132496.

Khoroshevsky, F., Zhou, K., Chemweno, S., Edan, Y., Bar-Hillel, A., Hadar, O., Rewald, B., Baykalov, P., Ephrath, J.E., Lazarovitch, N., 2024. Automatic root length estimation from images acquired in situ without segmentation. Plant Phenomics 6, 132. https://doi.org/10.34133/plantphenomics.0132.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324. https://doi.org/10.1109/5.726791.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125. 10.1109/CVPR.2017.106.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988. 10.1109/ICCV.2017.324.

Liu, Z., Marella, C.B., Hartmann, A., Hajirezaei, M.R., von Wirén, N., 2019. An age-dependent sequence of physiological processes defines developmental root senescence. Plant Physiol. 181 (3), 993–1007. https://doi.org/10.1104/pp.19.00809.

Lupo, Y., Schlisser, A., Dong, S., Rachmilevitch, S., Fait, A., Lazarovitch, N., 2022. Root system response to salt stress in grapevines (Vitis spp.): A link between root structure and salt exclusion. Plant Sci. 325, 111460. https://doi.org/10.1016/j.plantsci.2022.111460.

McCormack, M.L., Dickie, I.A., Eissenstat, D.M., Fahey, T.J., Fernandez, C.W., Guo, D., Helmisaari, H.S., Hobbie, E.A., Iversen, C.M., Jackson, R.B., Leppälammi-Kujansuu, J., 2015. Redefining fine roots improves understanding of below-ground contributions to terrestrial biosphere processes. New Phytol. 207 (3), 505–518. https://doi.org/10.1111/nph.13363.

Pierret, A., Moran, C.J., Doussan, C., 2005. Conventional detection methodology is limiting our ability to understand the roles and functions of fine roots. New Phytol. 166 (3), 967–980. https://doi.org/10.1111/j.1469-8137.2005.01389.x.

Prieto, I., Stokes, A., Roumet, C., 2016. Root functional parameters predict fine root decomposability at the community level. J. Ecol. 104 (3), 725–733. https://doi.org/10.1111/1365-2745.12537.

Rewald, B., Ephrath, J.E., 2013. Minirhizotron techniques. In: Plant Roots: the Hidden Half. Fourth Edition, CRC Press, pp. 735–750.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, pp. 234–241. 10.1007/978-3-319-24574-4_28.

Seethepalli, A., Dhakal, K., Griffiths, M., Guo, H., Freschet, G.T., York, L.M., 2021. RhizoVision Explorer: Open-source software for root image analysis and measurement standardization. AoB Plants 13 (6), p.plab056. https://doi.org/10.1093/aobpla/plab056.

Smith, A.G., Han, E., Petersen, J., Olsen, N.A.F., Giese, C., Athmann, M., Dresbøll, D.B., Thorup-Kristensen, K., 2022. RootPainter: deep learning segmentation of biological images with corrective annotation. New Phytol. 236 (2), 774–791. https://doi.org/10.1111/nph.18387.

Soda, N., Ephrath, J.E., Dag, A., Beiersdorf, I., Presnov, E., Yermiyahu, U., Ben-Gal, A., 2017. Root growth dynamics of olive (Olea europaea L.) affected by irrigation induced salinity. Plant and Soil 411, 305–318. https://doi.org/10.1007/s11104-016-3032-9.

Song, Z., Zhao, T., Jin, J., 2023. Early identification of root damages caused by western corn rootworms using a minimally invasive root phenotyping robot—MISIRoot. Sensors 23 (13), 5995. https://doi.org/10.3390/s23135995.

Sood, S., Singh, H., 2021. Computer vision and machine learning based approaches for food security: A review. Multimed. Tools Appl. 80 (18), 27973–27999. https://doi.org/10.1007/s11042-021-11036-2.

Tracy, S.R., Nagel, K.A., Postma, J.A., Fassbender, H., Wasson, A., Watt, M., 2020. Crop improvement from phenotyping roots: highlights reveal expanding opportunities. Trends Plant Sci. 25 (1), 105–118. https://doi.org/10.1016/j.tplants.2019.10.015.

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

Wu, Q., Pagès, L., Wu, J., 2016. Relationships between root diameter, root length and root branching along lateral roots in adult, field-grown maize. Ann. Bot. 117 (3), 379–390. https://doi.org/10.1093/aob/mcv185.

Yu, G., Zare, A., Xu, W., Matamala, R., Reyes-Cabrera, J., Fritschi, F.B., Juenger, T.E., 2020. Weakly supervised minirhizotron image segmentation with mil-cam. In: European Conference on Computer Vision. Springer International Publishing, Cham, pp. 433–449. 10.1007/978-3-030-65414-6_30.

Zhang, X., Wang, W., 2015. The decomposition of fine and coarse roots: their global patterns and controlling factors. Sci. Rep. 5 (1), 9940. https://doi.org/10.1038/srep09940.

Zhou, K., Jerszurki, D., Sadka, A., Shlizerman, L., Rachmilevitch, S., Ephrath, J., 2018. Effects of photoselective netting on root growth and development of young grafted orange trees under semi-arid climate. Sci. Hortic. 238, 272–280. https://doi.org/10.1016/j.scienta.2018.04.054.

Zhu, J., Ingram, P.A., Benfey, P.N., Elich, T., 2011. From lab to field, new approaches to phenotyping root system architecture. Curr. Opin. Plant Biol. 14 (3), 310–317. https://doi.org/10.1016/j.pbi.2011.03.020.