Expert Systems With Applications 299 (2026) 129930

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Rootex 2.0: Multi-head deep learning and graph-based analysis for automated barley root phenotyping

Maichol Dadi, Annalisa Franco, Alessandra Lumini *

Dept. of Computer Science and Engineering, University of Bologna, Via dell'Università 50, 47522, Cesena, FC, Italy

ARTICLE INFO

Keywords:
Root phenotyping
Barley root extraction
Automated root segmentation
Deep neural networks root system architecture
Plant phenotyping

ABSTRACT

Understanding plant root architecture under diverse environmental conditions is crucial for improving crop resilience and ensuring global food security. We present a fully automated method for segmenting barley root systems from high-resolution images and detecting keypoints such as tips and sources with high precision. At the core of our approach is *DeepRoot-3H*, a novel multi-head deep network built upon the DeepLabv3+ backbone, designed to jointly handle root segmentation and keypoint detection within a unified architecture. This integrated design enhances both the consistency and robustness of the outputs.

A dedicated post-processing stage further refines keypoint localization, effectively handling challenges such as dense root clusters and variability in image quality. The resulting predictions are then structured into a graph representation, on which a path-walking algorithm identifies biologically meaningful connections between tips and sources. This enables the generation of RSML files and the extraction of critical morphological traits.

To evaluate the system, we employ IoU and Dice scores for segmentation quality, alongside Euclidean and weighted distance metrics for tip and source detection. We also assess the biological consistency of the extracted traits—such as total root length, tortuosity, covered area, and outer angles—through correlation and discrepancy measures. Experimental results on a challenging benchmark dataset demonstrate significant improvements over existing techniques, confirming the effectiveness and reliability of our method for high-fidelity root system analysis.

1. Introduction

Efficient plant phenotyping is essential for advancing agricultural research, as it provides vital insights into how crops respond to environmental challenges, pests, and diseases. Among the various traits of interest, the structure and dynamics of root systems are particularly significant, influencing a plant's ability to access water and nutrients, tolerate abiotic stresses, and ultimately enhance yield stability. A precise and reliable characterization of root system architecture (RSA) can therefore guide breeding strategies that target more resilient and resource-efficient plant varieties, contributing to sustainable agriculture and global food security.

Despite this importance, accurately extracting and analyzing roots from large, high-resolution image datasets remains a complex endeavor. Variations in root architectures, heterogeneous growth mediums, non-uniform lighting, and overlapping root structures complicate segmentation and feature extraction. Early computational methods relied on heuristic approaches with limited generalization capabilities. More advanced techniques have integrated machine learning and deep learning frameworks to improve segmentation accuracy and scalability; however, existing models often struggle with the fine localization of critical features, such as tips and source points, and may require extensive manual intervention or specialized training sets.

To address these challenges, we propose *RootEx 2.0*, a significantly enhanced version of our automated root analysis pipeline (Dadi et al., 2025, 2024b). Drawing upon insights from previous iterations and state-of-the-art research, this approach adopts *DeepRoot-3H*, a novel multi-head deep learning architecture based on ResNet and Deeplabv3+. Unlike earlier solutions that focused primarily on root segmentation (Dadi et al., 2025, 2024b), *RootEx 2.0* incorporates dedicated heads for the segmentation of roots, tips, and sources. Class-specific weights help address imbalance issues, ensuring that smaller yet biologically significant structures are accurately identified. After segmentation and keypoint detection, we apply a customized post-processing phase that refines feature localization by splitting closely clustered contours and mitigating other artifacts. Finally, a path-walking algorithm traces the connections between detected tips and their corresponding sources, generating RSML

* Corresponding author.
E-mail addresses: maichol.dadi2@unibo.it (M. Dadi), annalisa.franco@unibo.it (A. Franco), alessandra.lumini@unibo.it (A. Lumini).

https://doi.org/10.1016/j.eswa.2025.129930
Received 5 June 2025; Received in revised form 9 September 2025; Accepted 1 October 2025
Available online 10 October 2025

files and enabling the extraction of a comprehensive set of morphological metrics.

The Root System Markup Language (RSML) (Lobet et al., 2015) is a standardized XML-based format for representing plant root system architectures, enabling structured storage, exchange, and quantitative analysis of root traits across different studies and tools. Various methods have been developed to extract root structures in RSML format, which can be broadly categorized into traditional approaches and deep learning-based techniques, depending on their underlying principles. A key resource enabling this work is the TILLMore CDC dataset (Dadi et al., 2025), which provides a rich collection of high-resolution barley root images accompanied by RSML annotations generated through the *RootTracer* tool (Dadi et al., 2024a). The annotated RSML files capture essential details such as root origins, tips, and branching points, offering a robust ground truth for evaluating segmentation and detection performance. Notably, these data span multiple growth sessions with varying lighting conditions and backgrounds, ensuring a challenging benchmark for our model's robustness.

By integrating a refined multi-head network architecture, tailored post-processing techniques, and a final path-walking step, *RootEx 2.0* addresses longstanding limitations in root phenotyping pipelines. Comparative analyses with established methods (Dadi et al., 2024b; Gaggion et al., 2021; Yasrab et al., 2019) suggest enhanced accuracy, improved scalability, and greater adaptability to complex imaging scenarios. In doing so, *RootEx 2.0* provides a versatile and reliable framework that supports the selection of resilient plant varieties and promotes sustainable agricultural practices through deeper and more actionable phenotypic insights.

The contributions of this work are:

1. *DeepRoot-3H*, a method that leverages a multi-head DeepLabV3+ architecture to independently segment roots, detect tips, and localize sources in high-resolution barley root images.
2. A strategy to address class imbalance and sparse keypoint detection using tailored loss functions, class-specific weighting, and customized data augmentation techniques.
3. An investigation of post-processing techniques for enhancing segmentation precision and keypoint localization, including overlapping tip resolution and distance-based filtering.
4. An integration of techniques to construct a graph-based representation of the root system, enabling accurate path-walking, RSML generation, and extraction of morphological traits.

2. Related works

Automated root phenotyping has become a critical component in modern plant science, enabling large-scale and non-destructive analysis of root system architecture (RSA). Over the years, a wide range of computational methods have been developed to process root images and extract morphological traits. These approaches can be broadly categorized into traditional image processing techniques and more recent deep learning-based solutions. Below, we review representative methods from both categories, highlighting their strengths and limitations in the context of root segmentation and structure reconstruction.

2.1. Traditional approaches

Traditional approaches to root phenotyping predominantly rely on classical image processing and rule-based algorithms. These methods typically involve steps such as image thresholding, edge detection, skeletonization, and morphological operations to segment root structures and extract phenotypic traits. Tools like *ImageJ* (Schneider et al., 2012), *Fiji* (Schindelin et al., 2012), and *ArchiSimple* (Pagès et al., 2014) apply such techniques to generate simplified representations of root systems from 2D images, enabling trait measurement such as length, area, and branching patterns.

Several frameworks incorporate geometric modeling and path-based tracing algorithms. For instance, *SmartRoot* (Leitner et al., 2012) and *RootNav 1.8* (Pound et al., 2013) rely on semi-automated graph traversal over preprocessed images to construct plausible root trajectories. These tools often use heuristics to constrain root pathfinding, combined with user supervision to validate or correct the computed structures.

Some systems, such as *RhizoVision Explorer* (Seethepalli et al., 2021), employ intensity- and contrast-based segmentation pipelines, followed by region-based analysis to quantify root properties like length distribution and diameter profiles. While these techniques can be effective under controlled imaging conditions, they tend to degrade in performance when applied to noisy, low-contrast, or overlapping root systems.

Another relevant example is *RootTracer* (Dadi et al., 2024a), which adopts a spline-based geometric representation to model root trajectories. Even though manual annotation is involved, the underlying framework emphasizes precise structural modeling through curve fitting and interpolation, enabling high-fidelity RSML exports for downstream morphological analysis.

A more recent contribution is the *Anti-gravity Stem-Seeking (AGSS)* restoration algorithm (Mingxuan et al., 2022). This method combines classical image processing steps—including illumination correction, grayscale conversion, binarization, morphological filtering, and skeletonization—with dedicated rules for junction and endpoint extraction. The algorithm identifies breakpoints in the primary roots and reconnects them using Bézier curve fitting, thus repairing incomplete root structures. Compared to frameworks that rely on user-guided graph navigation, the AGSS approach provides a fully automated repair procedure that can be directly contrasted with graph-based navigation strategies. However, because it does not leverage deep neural networks, the accuracy of the initial segmentation is lower, which in turn necessitates more elaborate post-processing to obtain consistent reconstructions.

In summary, traditional root analysis methods predominantly employ handcrafted pipelines based on low-level visual features, predefined heuristics, and geometry-based representations. While these techniques are interpretable and relatively lightweight, they generally lack robustness when applied to complex or high-throughput phenotyping scenarios, motivating the shift toward data-driven and learning-based solutions.

2.2. Deep learning-based approaches

The advent of deep learning has led to significant strides in root image analysis, enabling more robust segmentation and feature extraction. Methods like *DL-RootAnatomy* (Wang et al., 2020) employ machine learning algorithms to identify anatomical traits in root cross-section images. Several deep learning frameworks have focused on refining root segmentation and representation. For instance, *RootNav 2.0* (Yasrab et al., 2019) replaces semi-automatic steps of Pound et al. (2013) with a deep convolutional neural network that localizes seeds and root tips via heatmap regression, then applies shortest-path algorithms to reconstruct full root architectures. However, its segmentation performance varies with dataset complexity and conditions, and in some cases can be suboptimal.

ChronoRoot (Gaggion et al., 2021) employs CNN-based segmentation on time-lapse image sequences, integrating temporal consistency to track bulb growth dynamics. Tools like *faRIA* (Narisetti et al., 2021) and *ITErRoot* (Seidenthal et al., 2022) utilize specialized CNN models and iterative architectures, respectively, to handle thin, branched roots. *TARSNet* (Bhugra et al., 2022) introduces a topological loss and residual learning strategies to achieve state-of-the-art segmentation, highlighting the potential of advanced deep learning approaches in phenotyping tasks.

However, despite the progress made by both traditional and deep learning-based tools, many still struggle to generalize effectively to new datasets that present significantly different conditions. In our case, the chosen dataset of barley roots introduces new challenges—such as fine

Fig. 1. Three examples of dataset images illustrating common challenges for automated segmentation: (left) root tips protruding beyond the absorbent paper and water traces; (center) overlapping tips, tangled root hairs, and textured background; (right) an image with significantly higher brightness than the rest of the dataset.

root hairs, multiple overlapping roots (including those from different plants), and environmental conditions distinct from those on which these models were trained. In our recent works, RootEx 1.0 (Dadi et al., 2025) and the enhanced versione RootEx 1.5 (Dadi et al., 2024b), we observed that this hybrid approach outperforms the aforementioned methods, particularly in high-resolution images with multiple overlapping plants. It enables the identification of roots, tips, and sources for each individual plant, providing a more comprehensive understanding of complex root systems. However, a key limitation of this method is the absence of dedicated segmentation for tips and sources, which are instead inferred from the skeleton extracted from the segmented root mask. This indirect strategy hinders the accurate detection of tips that overlap with other roots or those that do not correspond to terminal branches in the skeleton. As a result, these existing methods often fail to deliver satisfactory performance, highlighting the need for a more robust and adaptable solution.

3. Materials and methods

3.1. Dataset overview

The TILLMore CDC labeled dataset (Dadi et al., 2025) consists of 120 high-resolution images of barley root systems (*Hordeum vulgare*), at a resolution of about 1500×1500 pixels (300 dpi). All images were acquired under controlled experimental conditions: plants were grown inside transparent CD cases lined with absorbent paper and photographed using a Nikon D5600 camera in a growth chamber.

The experimental setup involved two distinct phases of image acquisition. A first image was captured after 48 hours to document the early stages of root growth. Subsequently, the cases were rotated by 90°, and a second image (labeled with "R" in the dataset) was acquired 24 hours later to monitor the advanced stage of root development. Following acquisition, each image was cropped to tightly frame the area occupied by the CD case, reducing irrelevant background content and image resolution. The dataset spans multiple acquisition sessions with slight variations in lighting and exposure settings, increasing variability and robustness for model training.

Each image includes a manually annotated root system, created using the RootTracer tool (Dadi et al., 2024a). Annotations are provided in RSML format. The dataset was already split into training and testing sets (89 and 31 images, respectively). Since no validation set was provided, we randomly selected 7 images from the training set to be used for validation and parameter tuning. The 31 testing images include a total of 80 individual plants to be analyzed, as some images contain multiple root systems. Crucially, the dataset's split was designed so that images of the same plant captured at different growth stages were allocated to the same set to prevent data leakage and maintain a fair evaluation.

Several characteristics of the dataset present challenges for automated root segmentation, particularly in the advanced growth stage images (labeled "R"):

Several characteristics of the dataset present challenges for automated root segmentation:

- **Background variability:** Different types of absorbent paper were used during acquisition, resulting in a range of textures (smooth, rough) and occasional yellow stains or water traces that may interfere with root detection.
- **Lighting variation:** A subset of images is noticeably brighter than others, which may impact the model's robustness across different acquisition conditions.
- **Root overlap:** Up to three germinated seeds per image can result in entangled root systems, increasing annotation complexity and making segmentation more difficult.
- **Boundary occlusions and tip protrusions:** In some cases, root tips extend beyond the absorbent paper, protruding into the empty or transparent areas outside the CD case. These regions often lack visual contrast or structure, making the tips difficult to detect and segment reliably.

These aspects contribute to the overall difficulty of the dataset and are illustrated in Fig. 1, which presents visual examples of typical segmentation challenges.

3.2. Methods

The proposed method delivers a fully automated pipeline for segmenting and analyzing barley root systems, integrating keypoint detection and topological reconstruction. Beginning from a single high-resolution input image, it progressively transforms the raw data into a structured representation suitable for RSML generation and morphological measurements. The subsequent sections detail the underlying network architecture used for segmentation purposes, the adopted post-processing procedures, and the strategy employed for path extraction. A single high-resolution image is first processed by a multi-head segmentation network, producing distinct layers for roots, tips, and sources. Following a post-processing stage that refines these segmented elements, a graph construction and path analysis module derives a structured representation of the root architecture. This final step enables RSML file generation and the extraction of morphological metrics, providing a comprehensive characterization of the root system (Fig. 2).

3.2.1. Segmentation network architecture

We propose *DeepRoot-3H*, a multi-head segmentation framework based on DeepLabV3+ (Chen et al., 2018a), a state-of-the-art deep learning model originally designed for semantic segmentation tasks.

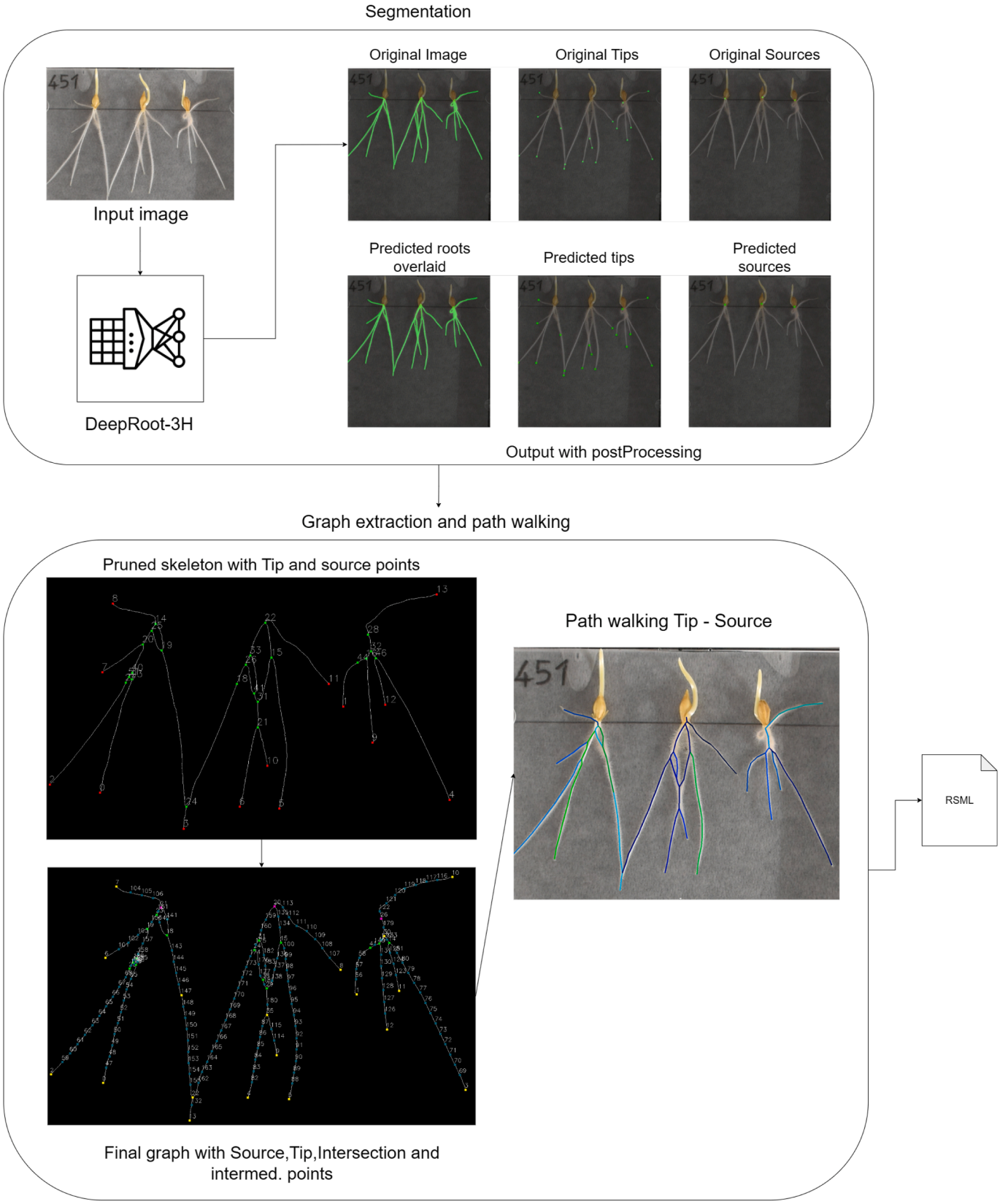


Fig. 2. High-level overview of the proposed method.

DeepLabV3+ was selected as the backbone architecture due to its strong performance in dense scene segmentation and its proven effectiveness in root phenotyping and related domains. Compared to other alternatives, it offers an excellent balance between accuracy, efficiency, and architectural flexibility, which makes it well-suited for our multi-head extension (see Section 4.1 for a quantitative comparison). In our configuration, the network is built upon a ResNet-50 backbone initialized with COCO-pretrained weights, ensuring robust feature extraction capabilities from the outset. To address the distinct challenges of root segmentation, tip localization, and source detection, we implement a multi-head design. Specifically, a shared encoder produces a common feature representation from the input image, which is then fed into three specialized decoders (heads). Each decoder targets a specific segmentation objective, generating a probability map for roots, tips, or sources. By isolating these tasks into separate output heads, the model can focus on refining features for each class independently, thereby improving accuracy and adaptability to complex root architectures.

The multi-head configuration enables flexible and targeted learning for each segmentation class. A standard multiclass segmentation approach, where a single decoder assigns each pixel to one of multiple classes, was not suitable for our task. This is because roots, tips, and sources are inherently different structures that do not conform to a mutually exclusive classification scheme. For instance, a pixel belonging to a root may simultaneously contribute to tip or source detection, making a hard assignment restrictive. In contrast, the multi-head design allows the network to learn independent yet complementary representations for each task, leading to more precise and semantically meaningful segmentation outputs. While the root head is dedicated to achieving precise boundary delineation and structural continuity, the tip head focuses on accurately identifying the positions of root tips, and the source head is optimized to locate key source points within the root system. This modular organization simplifies the optimization process, as each decoder can be fine-tuned using tailored loss functions, class weights, and post-processing techniques without negatively affecting the others. The resulting probability maps for roots, tips, and sources provide a structured representation of the segmented components. These maps serve as the foundation for subsequent graph construction and RSML generation steps.

3.2.2. Loss functions

The loss functions used in our model are designed to accommodate the distinct nature of each segmentation task. The final loss used for training is a weighted sum of the three individual loss terms:

$$\mathcal{L} = \alpha\mathcal{L}_{root} + \beta\mathcal{L}_{tip} + \gamma\mathcal{L}_{source} \tag{1}$$

where α, β, γ are hyperparameters controlling the contribution of each loss component. Their values were optimized via grid search.

For root segmentation, we employ a combination of Dice Loss and weighted Binary Cross-Entropy (BCE) Loss:

$$\mathcal{L}_{root} = \lambda_{root} \cdot \mathcal{L}_{Dice} + (1 - \lambda_{root}) \cdot \mathcal{L}_{BCE} \tag{2}$$

Dice Loss enhances contour precision:

$$\mathcal{L}_{Dice} = 1 - \frac{2 \cdot \sum_i p_i g_i}{\sum_i p_i^2 + \sum_i g_i^2 + \epsilon} \tag{3}$$

where $g_i \in \{0, 1\}$ is the ground-truth label, $p_i \in [0, 1]$ is the predicted probability (after sigmoid activation) and ϵ as a stabilization term.

Weighted Binary Cross-Entropy ($\mathcal{L}_{BCE}$) ensures classification stability and includes a weighting factor to address class imbalance by increasing the contribution of positive samples in the loss function:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_i \left(w \cdot g_i \log p_i + (1 - g_i) \log(1 - p_i) \right) \tag{4}$$

where, $w \geq 1$ is a coefficient that increases the loss associated with false negatives. This weighting is especially useful when the positive class is underrepresented. The value of w for each class is computed empirically based on the imbalance between positive and negative pixels:

$$w \approx \frac{N_{\text{neg}}}{N_{\text{pos}}} \tag{5}$$

Tip and source detection involve sparse and imbalanced data; therefore, we employ weighted Binary Cross-Entropy loss for both tasks. The weight w is independently computed for each task—root, tip, and source—based on the corresponding class imbalance in the training data.

$$\mathcal{L}_{tip} = \mathcal{L}_{source} = \mathcal{L}_{BCE} \tag{6}$$

3.2.3. Training set creation and data augmentation

To prepare the training set for semantic segmentation, we designed a preprocessing and augmentation pipeline aimed at enhancing model generalization, increasing data diversity, and addressing class imbalance.

Each high-resolution image was divided using a sliding window strategy, generating nine non-overlapping patches of size 512×512 pixels. This approach ensures full spatial coverage and allows the model to learn from localized root structures. Additionally, a total of 330 random crops of size 800×800 pixels were extracted across the dataset to introduce variability in both location and scale. These larger patches provide extended spatial context and help the network generalize across partial views and diverse background-foreground compositions.

To preserve global context, a subset of full-resolution images was also included in the training set. These images offer a holistic view of the root system, enabling the model to capture large-scale structural patterns that might be lost when using only cropped inputs.

All extracted patches, whether from fixed grids, random crops, or full-resolution images, were resized to a standard input size of 512×512 pixels to match the input requirements of the *DeepRoot-3H* network during training.

Given the class imbalance in the original dataset, where root classes were more prevalent than tips or sources, we applied simple data augmentation techniques to boost the presence of underrepresented classes. Specifically, images containing tip and source annotations were augmented through horizontal and/or vertical flipping. These operations produce mirrored versions of rare classes, improving their representation and variability within the training distribution.

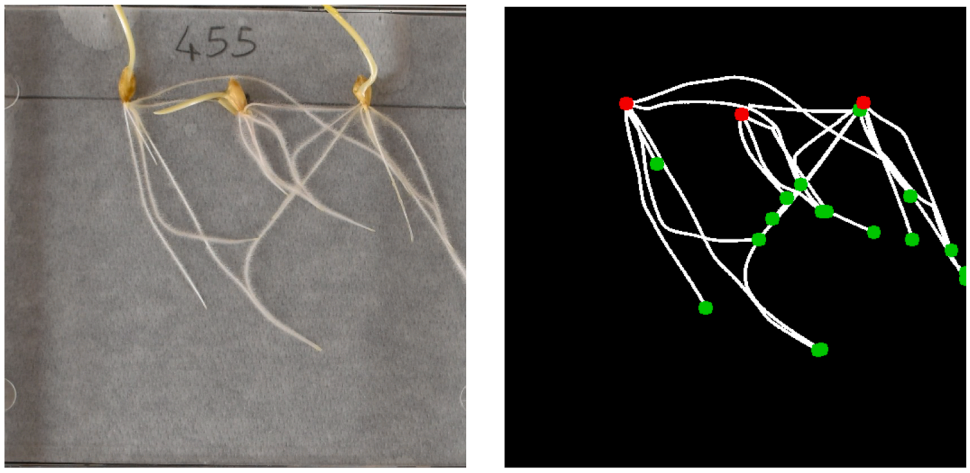
The final dataset composition, after applying data augmentation to the initial split, is as follows:

- **Training set:** 1440 images obtained through extensive augmentation of the original 82 training images (i.e., after excluding the 7 set aside for validation).
- **Validation set:** 150 augmented images derived from the 7 training images held out for validation, used for parameter tuning and intermediate model selection.

In addition to these targeted augmentations, a suite of general-purpose transformations (Buslaev et al., 2020) was applied to further increase robustness and model generalization. These include random flips, Gaussian blur, and additive Gaussian noise, simulating variations in imaging conditions. Input normalization was performed using ImageNet mean and standard deviation values to ensure consistent feature scaling.

The original RSML annotations were converted into three separate bitmap masks, each representing one of the semantic classes: roots, tips, and sources. Roots were rasterized using polylines with a width of 7 pixels, while tips and sources were encoded as filled circles with a fixed radius of $\sigma = 7.5$ pixels (at the original resolution).

To represent the root structures, dense binary masks were generated, highlighting the presence of root pixels across the image. In contrast, tips and sources—being discrete, localized points—were encoded as small filled regions centered on the annotated coordinates.



(a) Original image (b) Ground truth mask

Fig. 3. Qualitative visualization of the input image (a) and the combined semantic mask (b), where root (white), sources (red), and tips (green) are overlaid for illustrative purposes. In practice, these masks are generated as three separate binary maps.

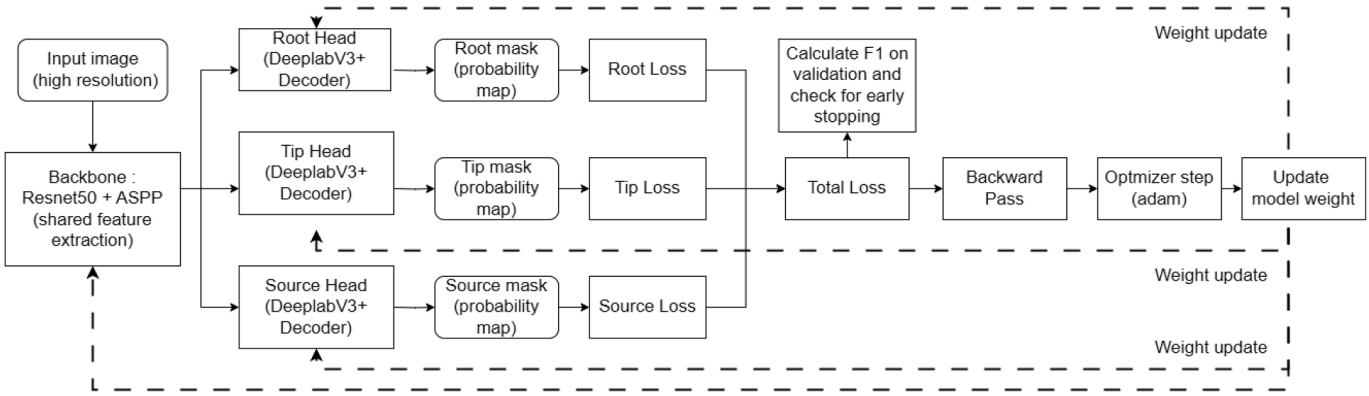


Fig. 4. Training pipeline of the multi-head *DeepRoot-3H* model.

This representation captures the spatial precision of such features while preserving their semantic identity, as shown in Fig. 3.

Each final mask encodes class-specific information at the pixel level, providing a rich and structured training signal for the segmentation heads of the *DeepRoot-3H* network. This design supports both fine-grained localization and broad structural understanding, critical for accurate multi-class semantic segmentation in root systems.

3.2.4. Training details

The main training loop, orchestrates the entire learning process (Fig. 4). It computes weighted losses for each class, ensuring that roots, tips, and sources are all optimized concurrently without overshadowing one another. Mixed precision training, enabled by `torch.amp`, reduces memory overhead and accelerates computations, making it feasible to train on large, high-resolution images. An early stopping criterion based on the sum of F1-scores helps prevent overfitting by halting training when no further improvements are observed. The input image is processed through a shared backbone and three aligned decoders (Root, Tip, Source). Weighted losses are calculated for each output, combined into the Total Loss $\mathcal{L}$, and used for gradient updates.

The training loop utilizes the Adam optimizer, chosen for its adaptability and robustness, with class-specific learning rates ($lr_{encoder}, lr_{root}, lr_{tip}, lr_{source}$) set according to difficulty and imbalance considerations. A learning rate scheduler such as ReduceLROnPlateau dynamically adjust the step size based on validation performance.

The training process explores multiple configurations for loss functions, optimizers, and learning rate schedulers. By systematically varying these hyperparameters, we identify optimal strategies that balance convergence speed, accuracy, and stability. Table 1 summarizes the explored components, the range of variations considered for each, and the best-performing setup identified through grid search. The final training configuration, including optimizer type, number of epochs, and batch size, is reported in Table 2.

3.2.5. Implementation and runtime details

The proposed pipeline was implemented in Python 3.12.5 using PyTorch 2.2.0 and CUDA 12.2. Data augmentations and image preprocessing were handled via the `Albumentations` library (v1.3.1), while graph construction and structural analysis relied on `NetworkX` (v3.2.1). All experiments were conducted on a Linux workstation equipped with an AMD Ryzen Threadripper PRO 5975WX 32-Core CPU (64 threads), 503 GB of RAM, and an NVIDIA RTX A6000 GPU (48 GB VRAM), running Ubuntu 20.04.6 LTS. The total processing time of the full pipeline—from semantic segmentation to RSML file generation—varied depending on the structural complexity of the root system. On average, the end-to-end runtime ranged between **10 and 20 seconds per image**, which includes segmentation, post-processing, graph construction, and morphological trait extraction. Although inference of the segmentation network alone is relatively fast (typically under **0.2 seconds**), subsequent stages—especially graph-based reconstruction and root trait computation—represent the dominant portion of the total runtime. These timings confirm that the method is suitable for high-throughput phenotyping pipelines while preserving structural accuracy and interpretability.

Table 1
Summary of the variations and changes applied to loss functions, learning rates, and scheduler parameters during the model optimization process. All combinations of the listed changes were evaluated. The last column reports the best-performing configuration.

Component	Range	Best
Loss		
λ_{root}	{0.3, 0.5, 0.7}	0.5
w_{root}	{1,2,5}	1
w_{tip}	{2, 5, 20}	20
w_{source}	{2, 5, 20, 60}	60
α	{1}	1
β	{2, 8, 10}	8
γ	{2, 8, 10, 20}	20
Learning Rates		
$lr_{encoder}$	{1e−4, 5e−5}	1e−4
lr_{root}	{1e−4, 5e−5}	1e−4
lr_{tip}	{1e−4, 3e−4, 5e−5}	1e−4
lr_{source}	{1e−4, 3e−4, 5e−5}	1e−4
Learning Rate Scheduler (ReduceLROnPlateau)		
Factor	{0.25, 0.5}	0.5
Patience	{5, 10, 15, 25}	10
Threshold	{0.001}	0.001
Minimum LR	{1e−6, 1e−7}	1e−6

Table 2
DeepRoot-3H experimental setup details.

Terms	Setup
Framework	PyTorch
Language	Python 3.12.5
Optimizer	Adam
Epochs	250
Batch size	20

3.3. Post-processing of network outputs

The outputs generated by *DeepRoot-3H* are subjected to a rigorous post-processing pipeline (Fig. 5) to refine segmentation masks and localize keypoints for tips and sources. This post-processing ensures that the predictions are accurate and robust, enabling reliable downstream analysis such as graph construction and RSML generation.

A fundamental step in this process is the binarization of the network's probabilistic outputs. Each output head (roots, tips, and sources) produces a dense map with values in the range $[0, 1]$, representing the confidence of each pixel belonging to the corresponding class. These maps are thresholded using class-specific parameters: τ_r for roots, τ_t for tips, and τ_s for sources. Thresholding serves to isolate confident predictions and suppress background noise.

The chosen threshold values reflect the relative certainty required to validate detections for each class. A lower threshold for sources accounts for their subtle appearance and low contrast, whereas a slightly stricter threshold for roots and tips helps preserve spatial precision.

The post-processing pipeline relies on several configurable parameters to optimize the output refinement. These parameters are summarized in Table 3. All parameters were optimized on the validation set to maximize overall performance.

Keypoint detection for tips and sources is a critical step in the post-processing pipeline. This procedure ensures precise localization of these biologically significant features, even in challenging conditions such as dense root clusters or overlapping structures. The following steps outline the detection process:

1. **Contour detection:** Binary masks for tips and sources are analyzed using contour-finding algorithms to identify individual regions representing potential keypoints.
2. **Centroid calculation:** For each detected contour, the center of mass is calculated to approximate the location of the keypoint. Alternatively, a minimum enclosing circle is fitted to handle irregularly shaped regions, ensuring robustness in diverse scenarios.
3. **Detection of overlapping tips:** Contours with an area greater than λ_{area} are assumed to contain multiple overlapping tips. In such cases:
   - To estimate the number of tips within the region, the total contour area is divided by the area of a single tip, which is approximated using an enlarged radius of $\frac{4\sigma}{3}$.
   - These tips are then evenly distributed within the contour using the same enlarged radius, ensuring that overlapping tips are correctly localized.
4. **Detection of missing tips:** For each segmented root, if no detected tip falls within its boundaries, the root is assumed to be missing a tip. In such cases, the tip is automatically assigned to the external node located at the end of the root path, ensuring completeness of the graph structure.

3.4. Graph creation and processing

Following the generation of a post-processed root mask(Fig. 6a), we create a topological graph of the root system to enable structured path analysis. This entire procedure consists of two main phases: (i)*skeletonization and basic pruning* to produce a clean initial graph (Fig. 6b), and (ii) *advanced graph refinement* steps to ensure consistent spacing, correct alignment of tip/source keypoints, and removal of spurious subgraphs(Fig. 6c). The outcome is a graph whose nodes represent the tips, sources, and intersection points of the root architecture, which is then ready for path-walking and RSML generation (Fig. 6d).

3.4.1. Skeletonization and basic pruning

We begin by applying a standard thinning technique to the binary root mask in order to reduce each connected root region to a one-pixel-wide centerline. This preserves the overall connectivity of the root architecture while discarding unnecessary thickness. Each nonzero pixel in the skeleton represents part of a potential edge in the topological graph. We then invoke a graph-construction routine which identifies two node types:

1. *External nodes*: Skeleton pixels with exactly one neighboring pixel (*degree*= 1), generally corresponding to tips or sources in the root.
2. *Intermediate nodes*: Pixels with exactly 2 neighbors.
3. *Intersection nodes*: Pixels with three or more neighbors (*degree*≥ 3), indicating a branching point.

The pixels lying between consecutive nodes are grouped into *edges*, each storing its path (the pixel coordinates) and the corresponding length. In this way, we obtain an initial multi-graph G that preserves the full topology of the root skeleton.

Skeletonization may produce noisy or fragmentary branches that terminate in external nodes. To remove them, we apply a threshold-based pruning step which iteratively discards any external node and its incident edge if that edge is shorter than a specified minimum length λ_a. This ensures that minor or spurious stubs do not remain in the final structure.

After the pruning step, the degree of some nodes may decrease—for instance, a node originally of degree 3 may become degree 2. In such cases, the intersection no longer represents a genuine branching point but rather acts as a simple pass-through. To address this, a merging routine removes the node and directly connects its two neighbors with a single edge. This step helps reduce unnecessary fragmentation in the graph.

Table 3
Post-processing parameters.

Parameter	Value	Description
τ_r, τ_t	0.5	Roots and tips binarization thresholds.
τ_s	0.3	Sources binarization threshold.
σ	7.5 px	GT keypoints radius and normalization factor representing the expected spatial resolution of the keypoints.
λ_{area}	$40 \cdot \sigma$	Minimum area (in pixels) used to detect overlapping tips. Contours exceeding this area are split into multiple tips based on the λ_{radius} parameter.
λ_t	5 px	Merging threshold for intersection nodes
λ_a	$8 \cdot \lambda_t$	Arc-length spacing for inserting intermediate nodes and path pruning lenght threshold.

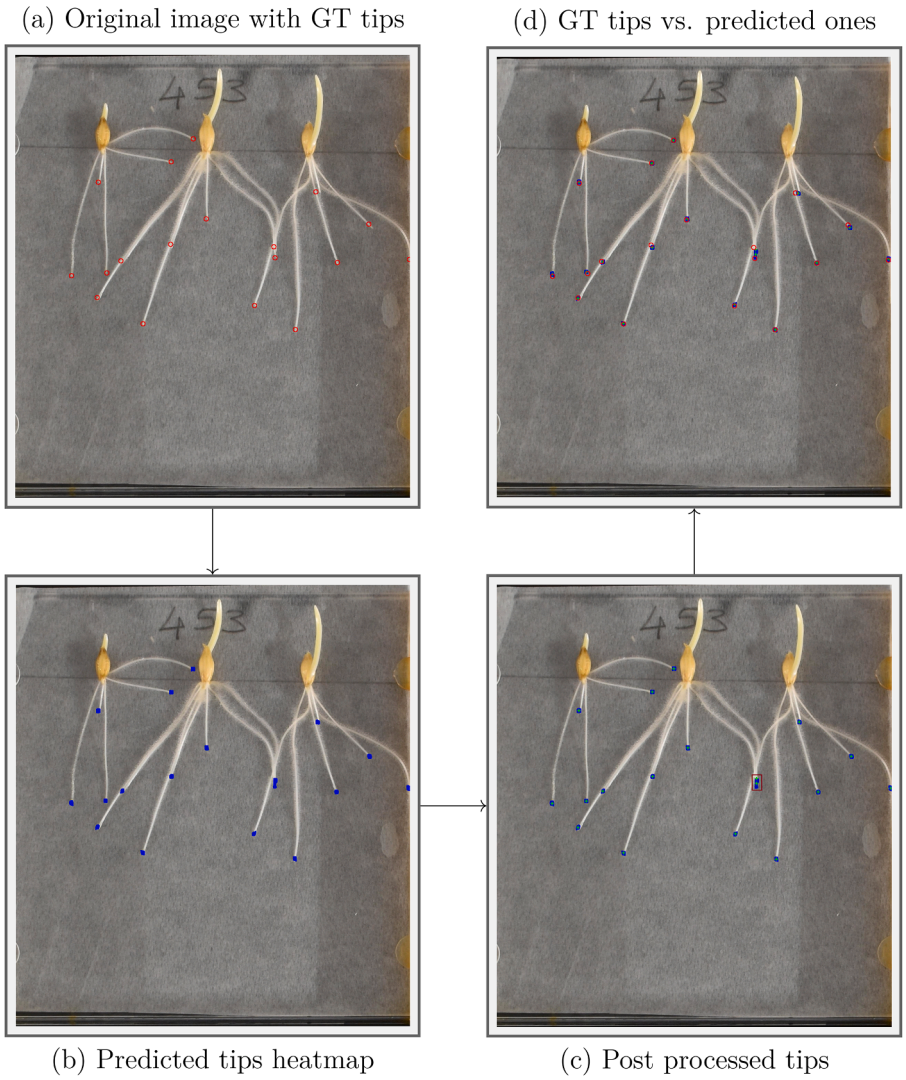


Fig. 5. Post-processing pipeline visualization. (a) Original image with ground truth tip annotations shown as red circles. (b) Overlay of predicted tips visualized as a blue mask. (c) Post-processed prediction where rounded red tips are separated into multiple distinct tips. (d) Final comparison highlighting the differences between ground truth tips (red circles) and the post-processed predicted tips (blue circles).

3.4.2. Advanced graph refinement

In cases where multiple intersection nodes lie within a small distance threshold λ_t, they are merged into a single node located at their centroid. This step mitigates artifacts introduced by skeletonization in dense root regions.

Following pruning and topological correction, additional intermediate nodes are inserted along each edge at regular arc-length intervals λ_a. This is done by traversing the skeleton coordinates and calculating the cumulative distance along each edge; new nodes are inserted such that no segment exceeds the predefined interval. In instances where two major nodes (e.g., intersections) are connected by a short edge without intermediate nodes, a midpoint node may be added to maintain structural consistency. This insertion strategy ensures an accurate piecewise-linear representation of root architecture, avoiding spline interpolation while remaining compliant with the RSML format, which encodes paths as sequences of straight-line segments. The resulting uniform node distribution enhances the robustness of path extraction and improves the reliability of morphological measurements.

Tip and source keypoints are concurrently refined to align with the graph structure. If a detected tip or source lies within a small radius of a graph node, it is snapped to that node. A tighter threshold λ_t is used for tips, while a slightly looser threshold $3\lambda_t$ is used for sources. These nodes are labeled accordingly as *tip* or *source* to preserve their biological role.

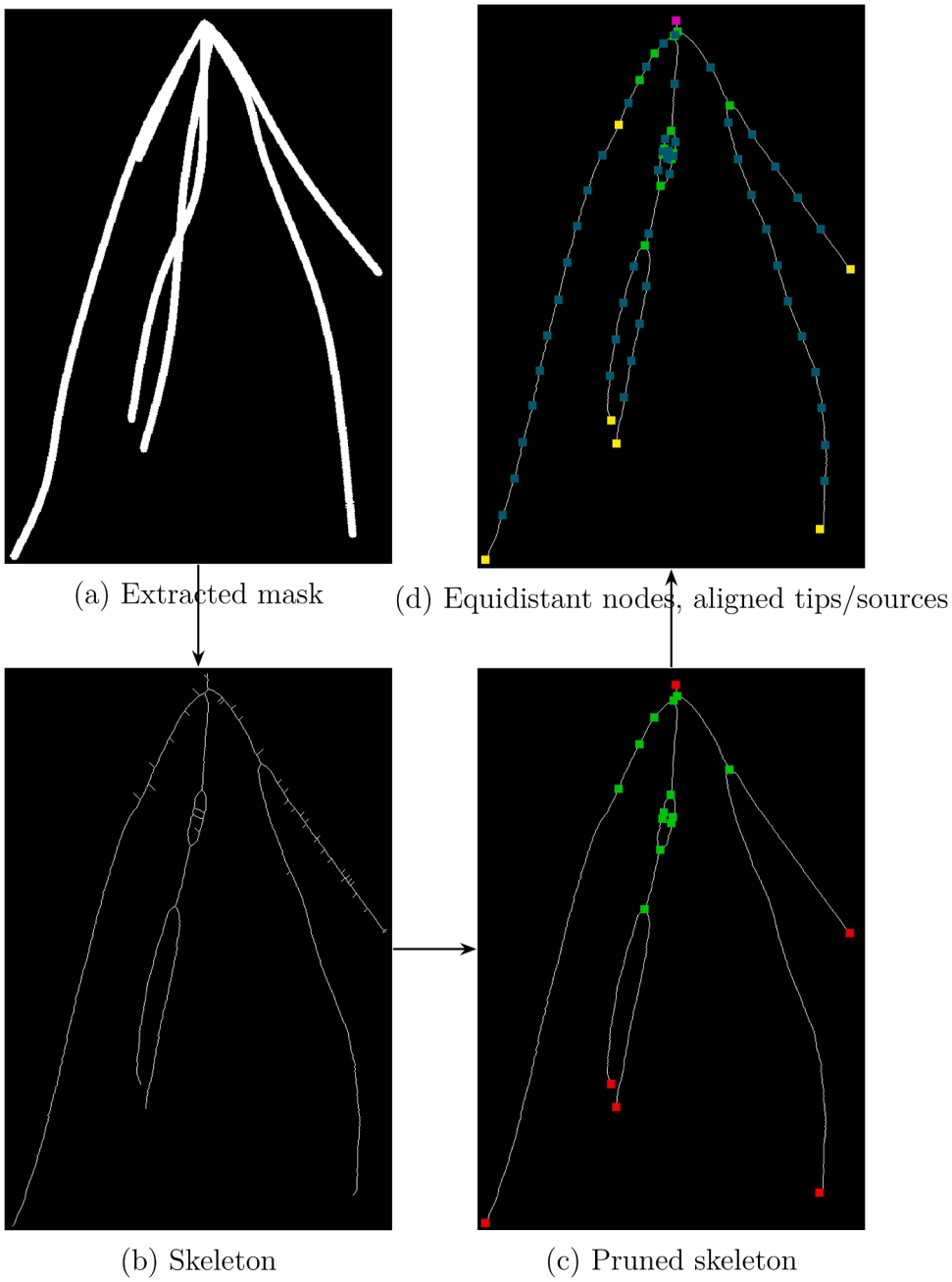


Fig. 6. Graph creation pipeline overview. (a) Initial binary mask from segmentation; (b) skeleton computed from the mask; (c) pruned skeleton with node type annotations (red = external, green = intersection); (d) final graph after equidistant node insertion (aqua), keypoint alignment (yellow = tips, pink = sources), and isolated node removal. Arrows indicate the processing flow.

Finally, any isolated subgraph not connected to at least one known source is removed. Each node is checked for connectivity to a source, and components lacking such a connection are discarded. This ensures that only the main root structure anchored at source nodes is retained.

3.5. Path walking and best-path selection

The path walking module is responsible for traversing the refined root graph from tip nodes toward source nodes, extracting all valid candidate paths, and subsequently selecting the optimal subset of these paths. This process is necessary because the input images often contain multiple roots growing in close proximity or intersecting, leading to a graph structure that deviates from the ideal tree-like topology required to accurately model all tip-to-source paths.

3.5.1. Valid path extraction

Given a graph $G = (V, E)$ derived from the binary plant image, valid paths are extracted by traversing the graph from each tip node ($t \in T$) toward source nodes ($s \in S$). A valid path originates at a tip node, contains no repeated nodes (i.e., no cycles), and terminates at a source node. At each step, neighboring nodes are considered only if the angle between the current direction and the overall direction from the tip to the source remains below a specified maximum threshold $\theta_{\max} = 25°$. This angular constraint helps preserve biologically plausible directions of root growth. When a source is reached, the current path is recorded as valid. If multiple neighbors satisfy the angular constraint, alternative paths are temporarily stashed for later exploration. When no valid neighbors remain, the algorithm ensures completeness by selecting the shortest path from the tip to the nearest available source.

The extracted paths are grouped into:

- **Multiple tips paths**: Paths associated with tips having multiple potential routes to sources.
- **Unique tips paths**: Direct, unambiguous paths from tips to sources.

The Algorithm 1 (*extractValidPaths*) takes as input a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, with some of the vertices labeled as tip nodes T or source nodes S, and an angle threshold $\theta_{\max}$ that restricts the selection of neighbors based on angle constraints.

Algorithm 1 extractValidPaths.

Input: graph $G = (V, E)$, tips $T \subset V$, sources $S \subset V$, angle threshold $\theta_{\max}$
Output: set of valid paths $\mathcal{P}_v$
$\mathcal{P}_v \leftarrow \emptyset$ ▷ Initialize the set of valid paths
for $t \in T$ **do**
  DFS(t, $[t]$) ▷ Start DFS from each tip
  if $t \notin \bigcup_{P \in \mathcal{P}_v} P[0]$ **then** ▷ If no path has been inserted for this tip
    $\mathcal{P}_v.append(shortestPath(t, S))$ ▷ Add the shortest path
  end if
end for
return $\mathcal{P}_v$
procedure DFS(v, P)
  if $v \in S$ **then**
    $\mathcal{P}_v.append(P)$ ▷ Reached a source, save current path
    return
  end if
  $N \leftarrow neighbors(v, \theta_{\max})$ ▷ Select neighbors within angle threshold
  for $n \in N$ **do**
    if $n \notin P$ **then** ▷ Avoid cycles
      DFS(n, $P + [n]$)
    end if
  end for
end procedure

The output of the algorithm is a set of valid paths $\mathcal{P}_v$, which start from a tip node in T and end at a source node in S, ensuring that the paths are acyclic.

The algorithm performs a depth-first search (DFS) starting from each node in T to explore all possible paths to nodes in S. For each node, the algorithm constructs a path by recursively appending neighbors that satisfy the angle constraint.

The function *neighbors*$(v, \theta_{\max})$ returns the neighboring nodes of a given vertex v, constrained by the angle threshold θ_{max}. If a source node is reached during the search, the current path is added to $\mathcal{P}_v$ as a valid path.

This process continues until all possible valid paths from the tip nodes to the source nodes are explored and added to the output set $\mathcal{P}_v$. If no valid path is found starting from a tip (i.e., no path satisfies the tortuosity constraints), the shortest path from that tip is added instead (via the *shortestPath(t,S)* function).

3.5.2. Selection of optimal paths

After collecting all valid candidate paths in $\mathcal{P}_v$, a simulated annealing-based optimization procedure selects the best path to add to an already selected set $\mathcal{P}_0$ of paths by minimizing a cost function that integrates three factors, i.e. overlap, tortuosity and coverage:

$$C(\mathcal{P}_0, P_k) = \lambda_1 \, \mathrm{IoU}(\mathcal{P}_0, P_k) + \lambda_2 \, \tau(P_k) + \lambda_3 \, \mathrm{NN}(\mathcal{P}_0, P_k). \tag{7}$$

Intersection over union (IoU): The IoU quantifies the overlap between the nodes of the candidate path P_k and the nodes already present in the set $\mathcal{P}_0$, ensuring unique structural contributions from the new path. Specifically, the IoU is defined as the ratio of the intersection of the nodes of P_k with the nodes in the existing set $\mathcal{P}_0$ to the union of all nodes in $\mathcal{P}_0 \cup P_k$:

$$\mathrm{IoU}(\mathcal{P}_0, P_k) = \frac{|\bigcup_{P \in \mathcal{P}_\Phi} P \cap P_k|}{|\bigcup_{P \in \mathcal{P}_0} P \cup P_k|} \tag{8}$$

Tortuosity (τ). Given a candidate path $P_k = \langle v_1, v_2, \ldots, v_n \rangle$ with tip v_1 and source v_n, let $\|v_i - v_j\|$ denote the Euclidean distance between nodes v_i and v_j, the tortuosity of P_k is defined as

$$\tau(P_k) = \frac{len(P_k)}{\|v_1 - v_n\|}, \qquad \tau(P_k) \geq 1. \tag{9}$$

where

$$len(P_k) = \sum_{i=1}^{n-1} \|v_i - v_{i+1}\| \tag{10}$$

is the length of a path. A perfectly straight path has $\tau = 1$; any deviation from the direct route increases τ proportionally to the extra distance travelled. This scalar therefore quantifies the overall curvature of the path and is used as a penalty term in the optimization cost.

Number of new nodes (NN): Coverage is measured by counting the nodes in P_k that are not already present in the set $\mathcal{P}_0$. This encourages the addition of new structural contributions:

$$\mathrm{NN}(\mathcal{P}_0, P_k) = |\{v \in P_k : v \notin \bigcup_{P \in \mathcal{P}_0} P\}| \tag{11}$$

The hyperparameters $\lambda_1, \lambda_2, \lambda_3$ control the relative importance of each factor in the cost function. Specifically, λ_1 penalizes overlap between paths via the Intersection over Union (IoU) term, ensuring structural uniqueness. λ_2 modulates the influence of tortuosity, enforcing spatial coherence along the path. λ_3 rewards structural novelty by favoring paths that introduce new nodes to the graph. In our experiments, we set $\lambda_1 = 2$, $\lambda_2 = 3$, and $\lambda_3 = 1$.

3.6. Evaluation metrics

A rigorous assessment of root system analysis methods requires well-defined and task-specific performance indicators. In this work, we evaluate our framework along three complementary dimensions: (i) segmentation accuracy at the pixel level, (ii) precision and completeness of tip and source detection, and (iii) structural fidelity of the reconstructed root system architecture. The following subsections introduce the metrics adopted for each of these aspects, together with the underlying matching and assignment procedures used to align predictions with the annotated ground truth.

3.6.1. Segmentation evaluation metrics

To rigorously evaluate the effectiveness of the proposed models on the root analysis task, we employ distinct performance metrics tailored to the nature of each subtask: root segmentation and keypoint (tip and source) detection. This section introduces the quantitative indicators adopted to assess the quality of the predictions with respect to the annotated ground truth.

Binary segmentation metrics. To evaluate the quality of binary segmentation masks, we adopt two standard region-based similarity metrics: the Intersection over Union (IoU) and the Dice coefficient. Let $\mathcal{M}_P$ be the predicted mask and $\mathcal{M}_G$ the corresponding ground-truth mask, both represented as binary sets of pixels. The metrics are defined as follows:

- **Intersection over union (IoU):** Also known as the Jaccard index, this metric measures the overlap between the predicted and ground-truth regions:

$$\mathrm{IoU} = \frac{|\mathcal{M}_P \cap \mathcal{M}_G|}{|\mathcal{M}_P \cup \mathcal{M}_G|} \tag{12}$$

- **Dice coefficient (Dice):** Equivalent to the F1 score for binary masks, the Dice coefficient gives more weight to correctly predicted pixels and is defined as:

$$\text{Dice} = \frac{2 \cdot |\mathcal{M}_P \cap \mathcal{M}_G|}{|\mathcal{M}_P| + |\mathcal{M}_G|} \tag{13}$$

Both metrics range in $[0, 1]$, with higher values indicating better segmentation accuracy. While IoU is more sensitive to boundary errors, Dice tends to be more robust when evaluating small objects.

Keypoint detection metrics. The evaluation of keypoint detection performance (for both tip and source categories) relies on a structured matching process, followed by error quantification. Given a set of predicted keypoints $\mathcal{K}_P = k_1, \dots, k_m$ and ground-truth keypoints $\mathcal{K}_G = g_1, \dots, g_n$, we first define a normalized pairwise distance matrix:

$$d_{ij} = \begin{cases} \dfrac{\|k_i - g_j\|}{\sigma}, & \|k_i - g_j\| \le \sigma, \\ 1, & \text{otherwise}, \end{cases} \tag{14}$$

where σ is a reference tolerance radius (see Table 3). This pairwise distance metric quantifies the similarity between predicted and ground-truth keypoints by computing their normalized Euclidean distances, capped at 1.0 for values exceeding the threshold σ. This design focuses the evaluation within a local radius, ensuring that only spatially plausible matches contribute to the score while heavily penalizing distant mismatches.

The optimal matching is obtained by solving a Linear Assignment Problem (LAP) over the cost matrix $[d_{ij}]$, where the numbers of predicted keypoints (m) and ground-truth keypoints (n) may differ. To account for this mismatch, dummy rows or columns with fixed unit cost are added, so that unmatched keypoints are penalized uniformly. Let $l = \max(m, n)$ denote the size of the augmented cost matrix. The optimal one-to-one assignment $\mathcal{A}^\star$ and its associated cost $C_{\mathcal{A}^\star}$ are obtained by solving the following Linear Assignment Problem:

$$\begin{aligned} \mathcal{A}^\star &= \arg\min_{x_{ij} \in \{0,1\}} \sum_{i=1}^{l} \sum_{j=1}^{l} d_{ij} x_{ij}, \\ C_{\mathcal{A}^\star} &= \frac{1}{l} \sum_{i=1}^{l} \sum_{j=1}^{l} d_{ij} x^\star_{ij}, \\ \text{s.t.} \quad & \sum_{j=1}^{l} x_{ij} = 1 \quad \forall i = 1, \dots, l, \\ & \sum_{i=1}^{l} x_{ij} = 1 \quad \forall j = 1, \dots, l, \\ & x_{ij} \in [0, 1] \quad \forall i, j. \end{aligned} \tag{15}$$

From the optimal assignment $\mathcal{A}^\star$, we first define n_{ass} as the number of predicted tips correctly matched to ground-truth tips within a distance threshold. The summation considers only tips lying sufficiently close to its corresponding ground-truth tip ($d_{ij} < 1$).

$$n_{\text{ass}} = \sum_{\substack{i,j \\ d_{ij} < 1}} x^\star_{ij} \tag{16}$$

Then we derive the following performance indicators:

- **Missed keypoint ratio, i.e. False negative rate (FNR):** Proportion of unmatched ground-truth keypoints relative to the total number of ground-truth keypoints:

$$\text{FNR} = \frac{n - n_{\text{ass}}}{n} \tag{17}$$

- **Overestimated keypoint ratio, i.e. False positive rate (FPR):** Proportion of unmatched predicted keypoints relative to the total number of predicted keypoints:

$$\text{FPR} = \frac{m - n_{\text{ass}}}{m} \tag{18}$$

- **Assignment cost** ($C_{\mathcal{A}^\star}$): The cost of the optimal assignment, as defined in Eq. 15, penalizes both localization inaccuracies and unmatched detections, using normalized distances clipped at 1.0 and a unit cost for dummy assignments. Unlike classical classification metrics such as precision, recall, or F1-score, $C_{\mathcal{A}^\star}$ jointly accounts for spatial accuracy and detection completeness, making it particularly suitable for tasks where both localization and presence of keypoints are critical.

3.6.2. Graph construction evaluation metrics

To evaluate the graph skeleton navigation algorithms used for RSA extraction, the RSAs generated by the two methods investigated in this study were compared with manually annotated ground truth data. The effectiveness of each method was assessed based on its ability to accurately recover root system architecture measurements.

The metrics used to analyze the structure of a plant's root system include the following root-level indicators (Judd et al., 2015) (Fig. 7a):

- Root length (RL)
- Root tortuosity (RT), defined as the average deviation of roots from a straight path (see Eq. 9)
- Deviation angle (DA), the angle between the initial direction of the root at the source and the vertical axis
- Tip angle (TA), the angle between the direction of the root tip and the vertical axis

The metrics used at the plant level are (Fig. 7b):

- Total and maximum root length (TL, ML)
- Covered area (CA), computed as the area of the convex hull encompassing the entire root system
- Average root tortuosity (AT), defined as the mean tortuosity across all roots within the plant
- Outer angle (OA), the angle formed between the leftmost and rightmost outer roots

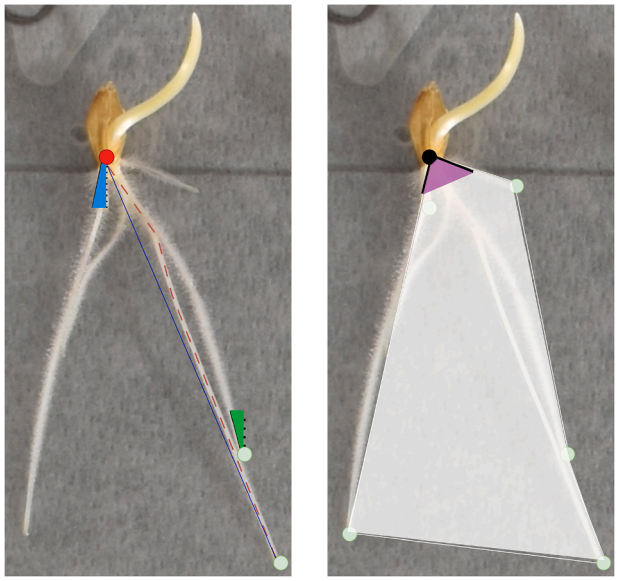
To quantify the agreement between the predicted and ground truth trait values, we computed Spearman's rank correlation coefficients (Yu et al., 2023), which assess the strength and direction of monotonic relationships without assuming linearity.

In addition, we evaluated performance using the Normalized Relative Discrepancy Index, which measures the relative difference between two values: one obtained from a specific method (A) and the other from the ground truth (G). This index is computed as the absolute difference between the two values, normalized by the ground truth value, and provides insight into the magnitude of the discrepancy relative to the true measurement: $\delta(M) = \frac{|M_A - M_G|}{M_G}$. However, for the DA and TA metrics, we used the absolute difference between the values (Δ), without normalization, to directly measure the magnitude of the difference between the predicted and true directions.

Plants are paired with their corresponding ground truth based on the position of their sources. To match the individual roots within each plant, we solve the following *Root Matching Problem*.

Root matching problem. The evaluation of root matching problem relies on a structured matching process, followed by error quantification. Given a set of predicted roots $\mathcal{R} = \{P_1, P_2, \dots, P_m\}$ and a set of GT roots $\mathcal{G} = \{G_1, G_2, \dots, G_n\}$ where each root P or G is represented as an ordered list of points.

We define the distance between couple of roots (P, G) by means of the Dynamic Time Warping (DTW) (Su et al., 2020) distance. Dynamic Time Warping (DTW) is a method for measuring similarity between two sequences that may vary in time or speed. In this study, DTW is employed to measure the similarity between two root polylines, normalized by the sum of their lengths (see Eq. 10).



a b

Fig. 7. The main Root System Measurements: (a) Root level measures: deviation angle (light blue), tip angle (green), root lenght (orange) and tip-source distance (blu); (b) Plant level measures: outer angle (purple) and convex hull(white).

$$DTW(P_i, G_j) = \frac{1}{len(P_i)+len(G_j)} \cdot \min_{W} \sum_{(l,k)\in W} \|p_l - g_k\| \ \forall i=1\dots m, j=1\dots n \tag{19}$$

where W is the a warping path aligning points between P_i and G_j.

The optimal matching among the sets of roots $\mathcal{R}$ and $\mathcal{G}$ is obtained by solving a Linear Assignment Problem (LAP) over the cost matrix $[DTW_{ij}]$, where the numbers of predicted roots (m) and ground-truth roots (n) may differ. To account for this mismatch, as done above for tips, dummy rows (columns) with fixed cost are added, so that unmatched roots are penalized. The cost of a possible unassigned root P is the minimum assignment to an already paired GT root G (or vice-versa if $m > n$):

$$DTW(P_k, G_j) = \min_{i=1\dots m} DTW(P_i, G_j) \quad \forall k = m+1\dots n, j = 1\dots n \tag{20}$$

The optimal one-to-one assignment $\mathcal{RA}^{\star}$ and its associated cost $C_{\mathcal{RA}^{\star}}$ are obtained by solving the Linear Assignment Problem as defined in Eq. 15 with cost matrix $[DTW_{ij}]$ instead of $[d_{ij}]$.

The cost of the optimal root-level assignment $C_{\mathcal{RA}^{\star}}$ quantifies the similarity between predicted and ground-truth roots using DTW-based distances. This cost penalizes both geometric discrepancies and unmatched roots, thereby providing a robust performance measure that accounts for structural accuracy and completeness.

4. Results

In this section, we provide a comprehensive evaluation of the proposed pipeline, focusing on two core components: the segmentation capabilities of the *DeepRoot-3H* network and the accuracy of the graph-based reconstruction of the root system architecture (RSA). The analysis is divided into two parts. First, we assess the quality of root segmentation and the localization of anatomical keypoints (tips and sources), using region-based metrics (Dice coefficient, IoU), detection error rates, and normalized spatial distance measures. We also evaluate the contribution of the proposed post-processing steps to the overall detection performance. Second, we examine the reliability of the path-walking algorithm and the graph construction procedure by comparing the extracted morphological traits against manually annotated ground truth, employing correlation coefficients, normalized discrepancies, and structural similarity measures such as Dynamic Time Warping (DTW).

Table 4
Comparison of binary root segmentation performance among widely used segmentation models on our test set.

Network	Dice	IoU	Time (s)
CafeNet (Wang et al., 2023)	0.722	0.57	2.50
HSNet (Li et al., 2021)	0.724	0.57	2.47
DeepLabV3+ (ResNet50) (Chen et al., 2018b)	0.74	0.58	2.21
UNet (ResNet50) (Ronneberger et al., 2015)	0.706	0.55	2.25

4.1. Comparison of backbone architectures for root segmentation

To justify the architectural choices underlying our system, we conducted a quantitative comparison of segmentation performance among several representative segmentation networks from the literature. The goal of this evaluation is twofold: (i) to position our model within the broader landscape of root segmentation methods, and (ii) to motivate the adoption of DeepLabv3+ as the backbone architecture for our multi-head framework. In this study, we compare four models: two well-established convolutional architectures, UNet (Ronneberger et al., 2015) and DeepLabv3+ (Chen et al., 2018b), which have been widely adopted in plant phenotyping tasks, and two more recent transformer-based networks, HSNet (Li et al., 2021) and CafeNet (Wang et al., 2023), originally developed for biomedical image analysis. All models were trained and evaluated on our annotated root image dataset under identical conditions to ensure a fair comparison. For consistency, UNet and DeepLabv3+ use the ResNet50 (He et al., 2016) backbone, while HSNet and CafeNet adopt the Pyramid Vision Transformer (PVTv2) (Wang et al., 2021) as their encoder.

As shown in Table 4, DeepLabv3+ achieves the best performance in terms of Dice and IoU among all evaluated architectures, while also maintaining competitive inference time. These results support its selection as the backbone for our multi-head architecture. Although DeepRoot-3H is trained for multi-class prediction (root, source, tip), this benchmark focuses solely on the binary segmentation of the *root* class to enable a fair comparison with single-class models.

4.2. Ablation study on segmentation and post-processing modules

This section presents a targeted ablation study aimed at quantifying the contribution of specific architectural and post-processing

Table 5
Performance comparison across different methods. Metrics are computed separately for root segmentation and keypoint detection (tip, source). Best results per column are highlighted in bold.

Method	IoU ↑			Dice ↑			FPR ↓		FNR ↓	$C_{\mathcal{A}^*}$ ↓
	Root	Tip	Source	Root	Tip	Source	Tip	Source	Tip	Tip
DL	0.58	0.51	0.45	0.74	0.63	0.6	**0.0121**	0.0132	0.0850	0.724
DL+	0.58	0.52	0.45	0.74	0.642	0.6	**0.0121**	0.0132	0.0752	0.72
CN	0.57	0.423	0.38	0.72	0.48	0.46	0.0300	0.0132	0.2100	2.21
DR	**0.61**	0.52	**0.48**	**0.766**	0.655	**0.612**	**0.0121**	0.0132	0.0510	0.71
DR+	**0.61**	**0.532**	**0.48**	**0.766**	**0.67**	**0.612**	**0.0121**	0.0132	**0.0437**	**0.704**

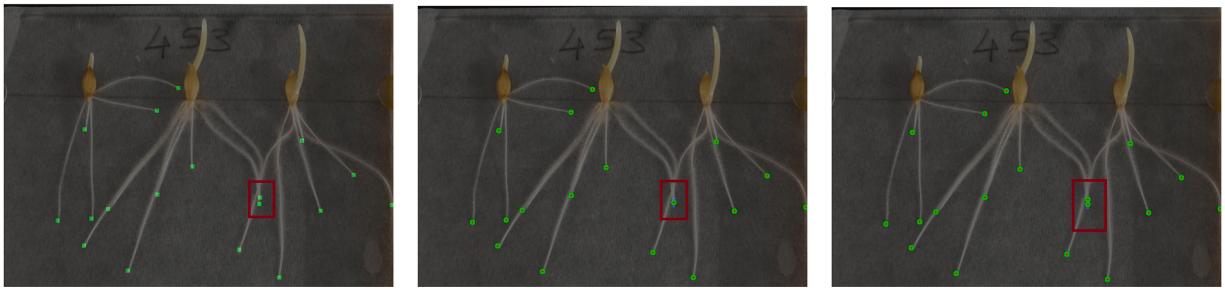


Fig. 8. Comparison of pre and post-processed outputs of *DeepRoot-3H*. Post-processing resolves overlapping keypoints and improves tip and source localization for accurate downstream analysis.

Table 6
Comparison of normalized discrepancy indices (δ ↓) and Spearman correlations (ρ ↑) for plant-level traits across methods. Best scores per column are highlighted in bold.

Method	δ ↓					ρ ↑				
	δ(TL)	δ(ML)	δ(AT)	δ(CA)	δ(OAD)	ρ(TL)	ρ(ML)	ρ(AT)	ρ(CA)	ρ(OAD)
RN2	19.52 %	7.16 %	**2.50 %**	8.65 %	25.02 %	0.757	0.884	0.821	0.866	0.647
RE1.5	20.55 %	5.81 %	5.24 %	5.78 %	22.67 %	0.837	0.913	0.645	0.890	0.608
RE2.0	**10.61 %**	**5.43 %**	**2.50 %**	**4.41 %**	**14.35 %**	**0.946**	**0.934**	**0.896**	**0.940**	**0.941**

components to the overall performance. We evaluate segmentation and keypoint detection under different model configurations, isolating the impact of (i) the post-processing module, and (ii) the multi-head architecture.

Specifically, we compare: (a) the baseline multiclass DeepLabV3+ model (DL); (b) the single-class CafeNet (Liu et al., 2024) used in *RootEx 1.5* (CN), where tips and sources are inferred post hoc (in a separate post-processing step); and (c) the proposed multi-head *DeepRoot-3H* architecture (DR). To assess the role of post-processing, both DL and DR are evaluated with and without the proposed refinement module (denoted as DL+ and DR+, respectively).

Performance is assessed using the metrics introduced in the previous section for root masks, as well as for tip and source keypoints. For keypoints, Dice and IoU are computed by converting detections into small binary masks centered on each keypoint location.

Results are summarized in Table 5, allowing for a comprehensive comparison in terms of segmentation quality, localization accuracy, and the robustness of detection. This evaluation highlights the benefits of the multi-head architecture and quantifies the improvements obtained through post-processing refinement. Overall, DR+ consistently delivers the best performance in all categories, validating the effectiveness of the multi-head design and the importance of integrating a tailored post-processing module for robust root system analysis.

In Fig. 8, qualitative results of the post-processing are shown: from the comparison between the DR and DR+ versions, it is evident that post-processing helps resolve issues arising from the network's inability to separate tips that are too close to each other. By analyzing the area size of the detected tips, it is possible to address this problem and obtain a result that aligns correctly with the ground truth (GT).

4.3. Graph construction: Quantitative results

This section presents a quantitative comparison of plant-level reconstruction accuracy across three methods: *RootNav 2.0* (RN2) (Yasrab et al., 2019), *RootEx 1.5* (RE1.5) (Dadi et al., 2024b), and our proposed *RootEX 2.0*(RE2.0).

The results presented in Table 6 highlight the significant improvements achieved by the proposed RE2.0 pipeline over the compared methods. In terms of discrepancy, RE2.0 demonstrates more accurate reconstruction of key plant-level traits, with particularly notable gains in the estimation of geometric and spatial properties. This is further supported by the consistently stronger rank correlations observed for RE2.0, which indicate a more faithful preservation of phenotypic variation across samples.

The method proves especially effective in capturing both global structural traits and localized features such as tip orientation, suggesting a well-balanced integration of precision and robustness. Unlike previous approaches, RE2.0 maintains phenotypic order with high reliability, underlining its utility for downstream quantitative analyses. The combined evidence suggests that RE2.0 not only reduces reconstruction error but also better reflects biologically meaningful variability across different plants.

The results reported in Table 7 provide a detailed evaluation of root-level reconstruction performance, highlighting the improvements

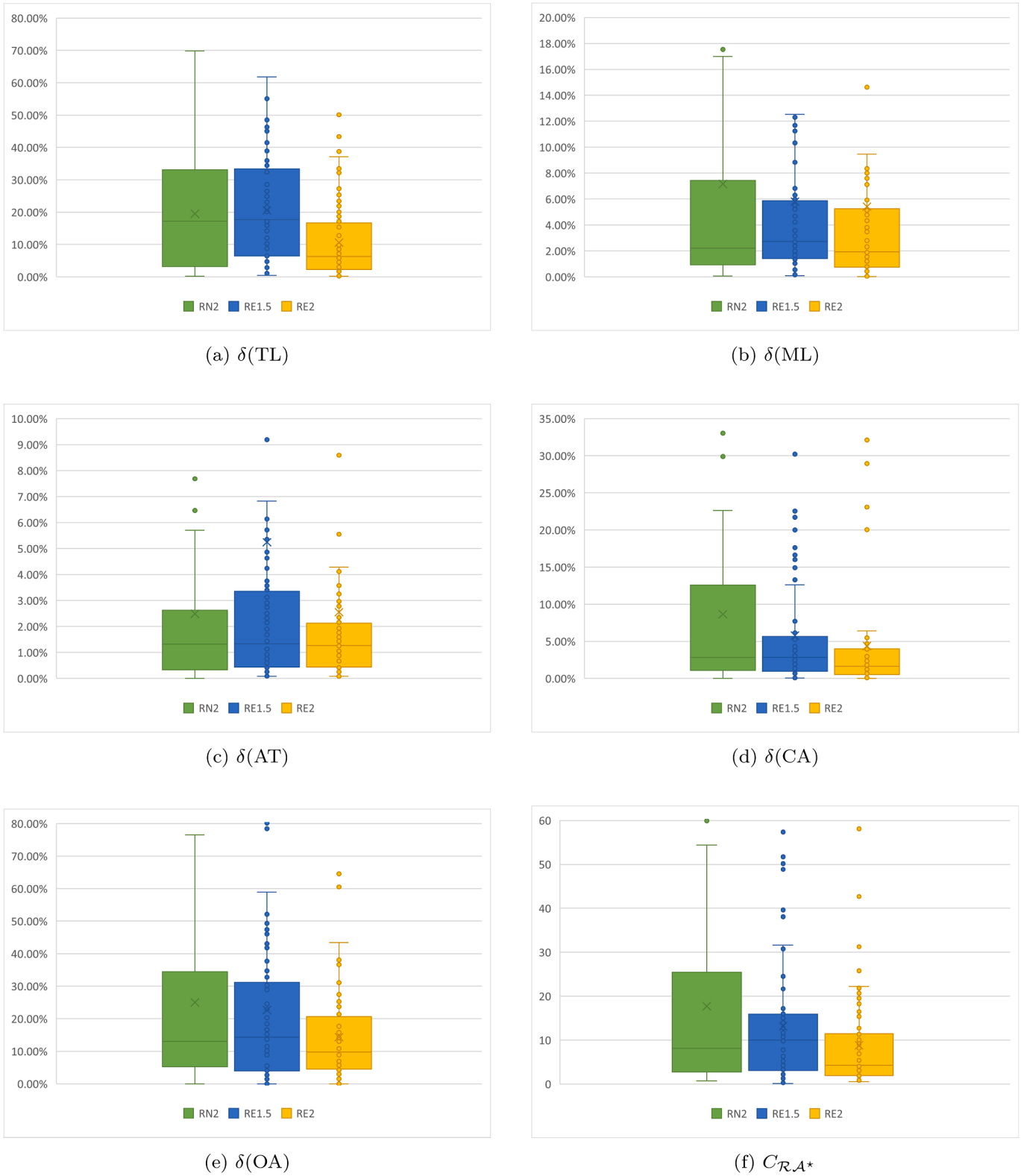


Fig. 9. Box plots of plant-level discrepancy indicators (δ(TL), δ(ML), δ(AT), δ(CA), δ(OA)) and root assignment cost ($C_{\mathcal{RA}^\star}$). Lower values indicate better performance.

introduced by RE2.0 across multiple metrics. Compared to both RN2 and RE1.5, the proposed method achieves consistently lower normalized discrepancies and angular errors, particularly in root length and tip angle. These reductions suggest a more precise capture of individual root geometries and orientations.

Furthermore, RE2.0 exhibits the highest Spearman correlations across all traits, indicating its strong ability to preserve the phenotypic ranking of roots. This consistency in trait ordering reinforces the method's reliability in capturing biologically relevant variation at the fine-grained level.

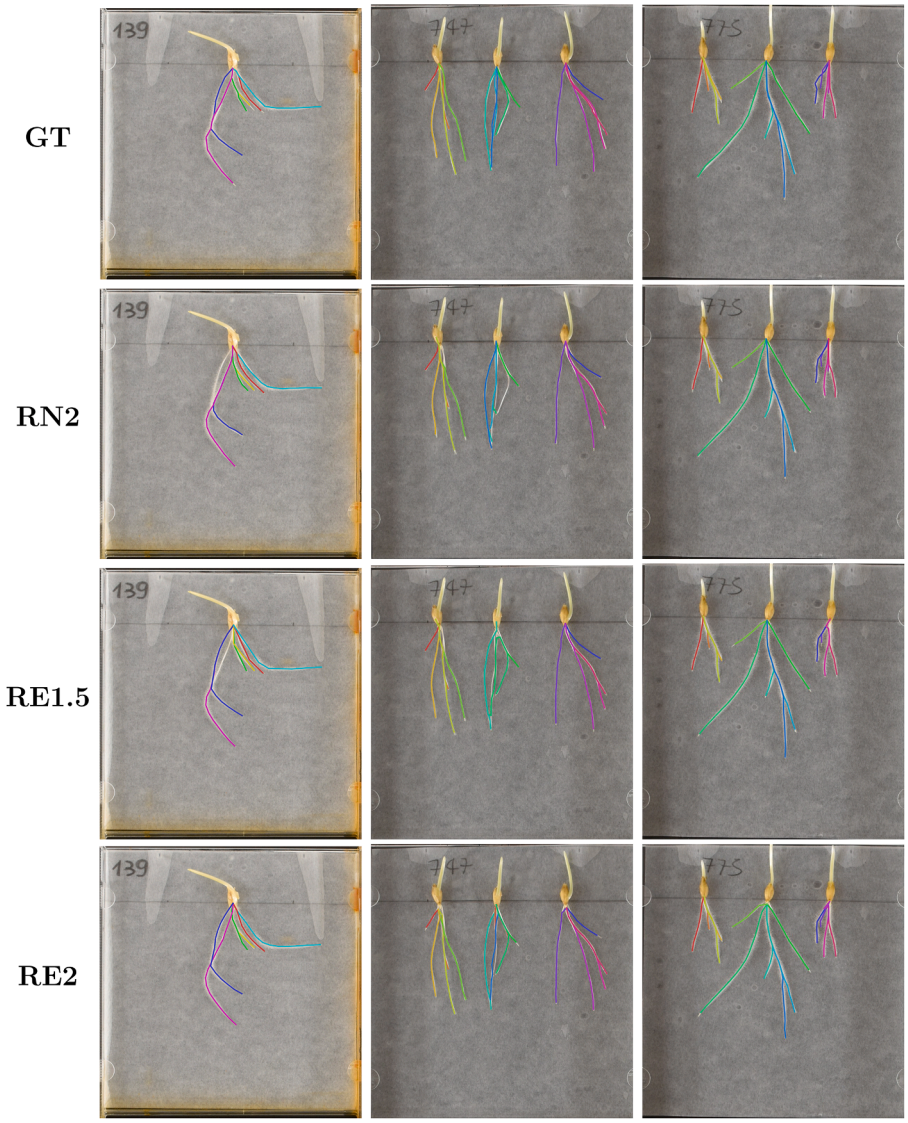


Fig. 10. Qualitative comparison for three samples of easy difficulty. Rows show the ground truth (GT) and predictions from **RN2**, **RE1.5** and **RE2**.

Table 7
Root-level performance metrics. Lower δ, Δ and higher ρ values are better. $C_{\mathcal{RA}^\star}$ is the cost of the root assignment problem and must be minimized.

Method	$\delta \downarrow$		$\Delta \downarrow$		$\rho \uparrow$				$C_{\mathcal{RA}^\star} \downarrow$
	δ(RL)	δ(RT)	Δ(DA)	Δ(TA)	ρ(RL)	ρ(RT)	ρ(DA)	ρ(TA)	
RN2	17.42 %	2.55 %	7.45	7.16	0.91	0.63	0.74	0.80	17.7
RE1.5	8.48 %	2.31 %	**6.19**	4.5	0.93	0.66	0.81	0.87	13.6
RE2.0	**8**%	**2.18**%	6.7	**4.1**	**0.94**	**0.68**	**0.88**	**0.92**	**9.3**

Most notably, RE2.0 minimizes the root assignment cost $C_{\mathcal{RA}^\star}$ more effectively than the baseline methods, confirming the overall quality and coherence of the predicted structures. This metric, which encapsulates both geometric alignment and completeness of the root matches, provides a comprehensive indicator of performance. Overall, these results demonstrate that RE2.0 yields more accurate, consistent, and biologically faithful root reconstructions than both traditional and earlier deep learning-based approaches. These quantitative findings are further supported by the boxplots in Fig. 9, which visualize the distribution of errors and assignment costs across methods. RE2.0 shows lower median values and reduced variance in all metrics, highlighting its robustness and consistency, especially in traits such as total length (TL), outer angle (OA), and covered area (CA).

5. Discussion

This section discusses the effectiveness and limitations of the proposed approach, with particular attention to the qualitative behavior of the graph-based root system reconstruction under varying levels of structural complexity. Here, root system complexity refers to the combined visual and topological challenges in the images, including dense root networks, overlapping segments, partial occlusions, and ambiguous structures near image borders.

To guide the analysis, we distinguish three representative levels of complexity: easy, medium, and hard. In the easy case, root systems are sparse, clearly separated, and free from significant visual interference, which facilitates accurate path extraction. Medium-complexity systems

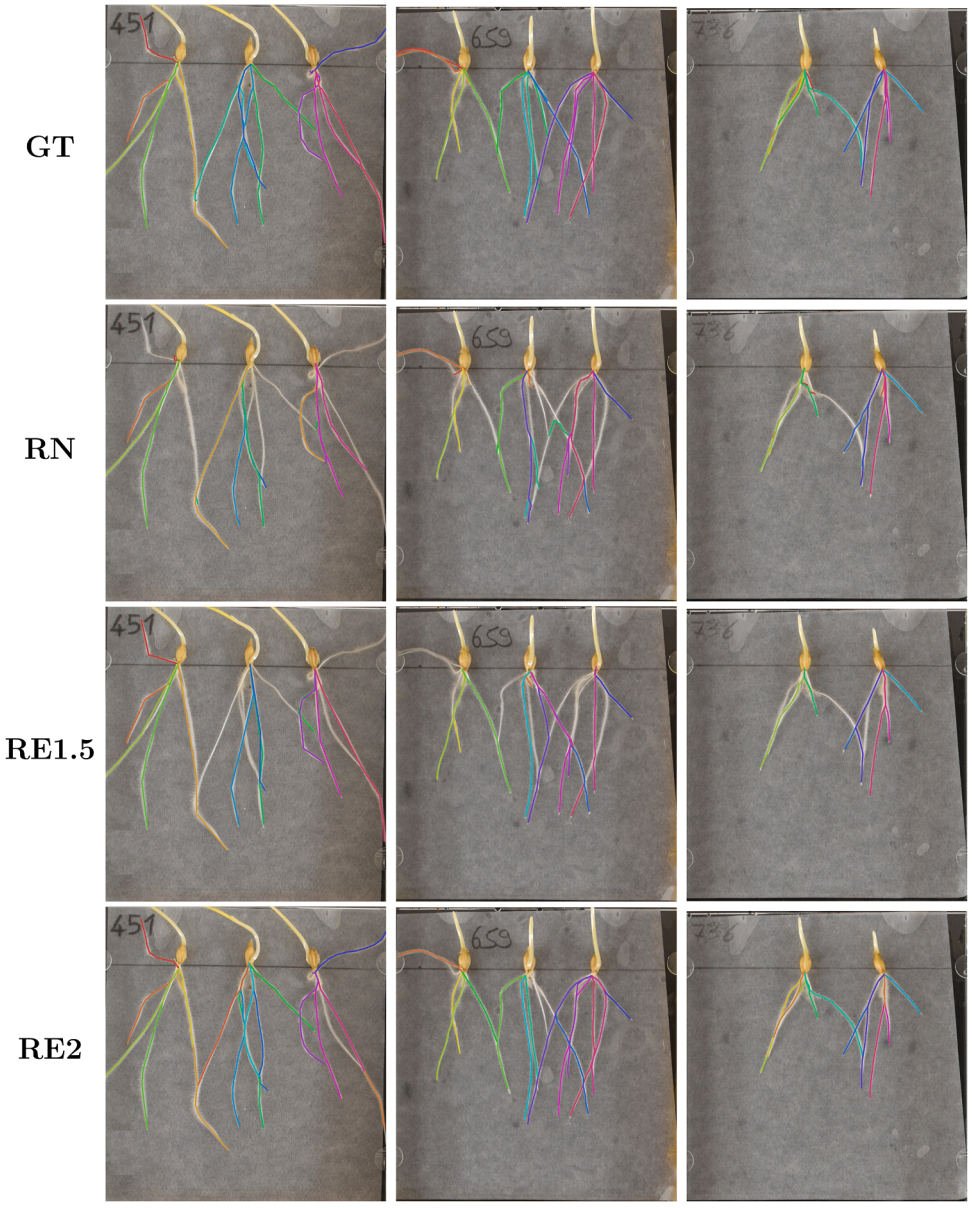


Fig. 11. Qualitative comparison for three samples of medium difficulty. rows show the ground truth (GT) and predictions from **RN2**, **RE1.5** and **RE2**.

Table 8
Average Normalized discrepancy indices (δ) across images of varying difficulty and methods. Best scores per *column* are highlighted in bold.

Easy Difficulty					
Method	δ(CA)↓	δ(TL)↓	δ(ML)↓	δ(RT)↓	δ(OAD)↓
RN2	2.52 %	2.82 %	3.80 %	1.12 %	25.56 %
RE1.5	2.58 %	9.74 %	2.73 %	2.62 %	27.01 %
RE2	**2.48 %**	**2.05 %**	**2.46 %**	**1.03 %**	**10.70 %**
Medium Difficulty					
RN2	10.95 %	32.88 %	9.91 %	3.03 %	53.70 %
RE1.5	13.09 %	30.16 %	**7.03 %**	3.55 %	47.83 %
RE2	**8.12 %**	**17.22 %**	8.48 %	**2.07 %**	**21.33 %**
Hard Difficulty					
RN2	17.66 %	**19.68 %**	14.60 %	7.33 %	35.24 %
RE1.5	14.04 %	38.30 %	5.15 %	7.35 %	51.50 %
RE2	**7.00 %**	19.99 %	**4.60 %**	**5.61 %**	**13.89 %**

exhibit moderate overlap, some background clutter, and uncertain root terminations near image boundaries, which increase the risk of incorrect reconstruction. In the hard case, root systems are densely packed with numerous intersections and occlusions, often exhibiting strong visual ambiguity that poses a significant challenge for both segmentation and graph inference.

Figs. 10, 11, and 12 provide representative examples for each complexity level. Each figure displays the ground truth root architecture alongside the predicted RSA generated by *RootNav 2.0* (RN2), *RootEx 1.5* (RE1.5), and the proposed *RootEx 2.0*(RE2.0). The reconstructions are color-coded according to the $\mathcal{A}^\star$ assignment among tips (see Section 3.6.1), ensuring consistent visual correspondence between predicted and ground truth root paths. These visualizations complement the quantitative evaluation by offering insight into the structural correctness and completeness of the reconstructed graphs. By progressively examining systems of increasing complexity, the comparison highlights the strengths and limitations of each method in recovering fine-grained topology and accurately detecting key elements such as tips and sources. A corresponding numerical comparison for the three representative

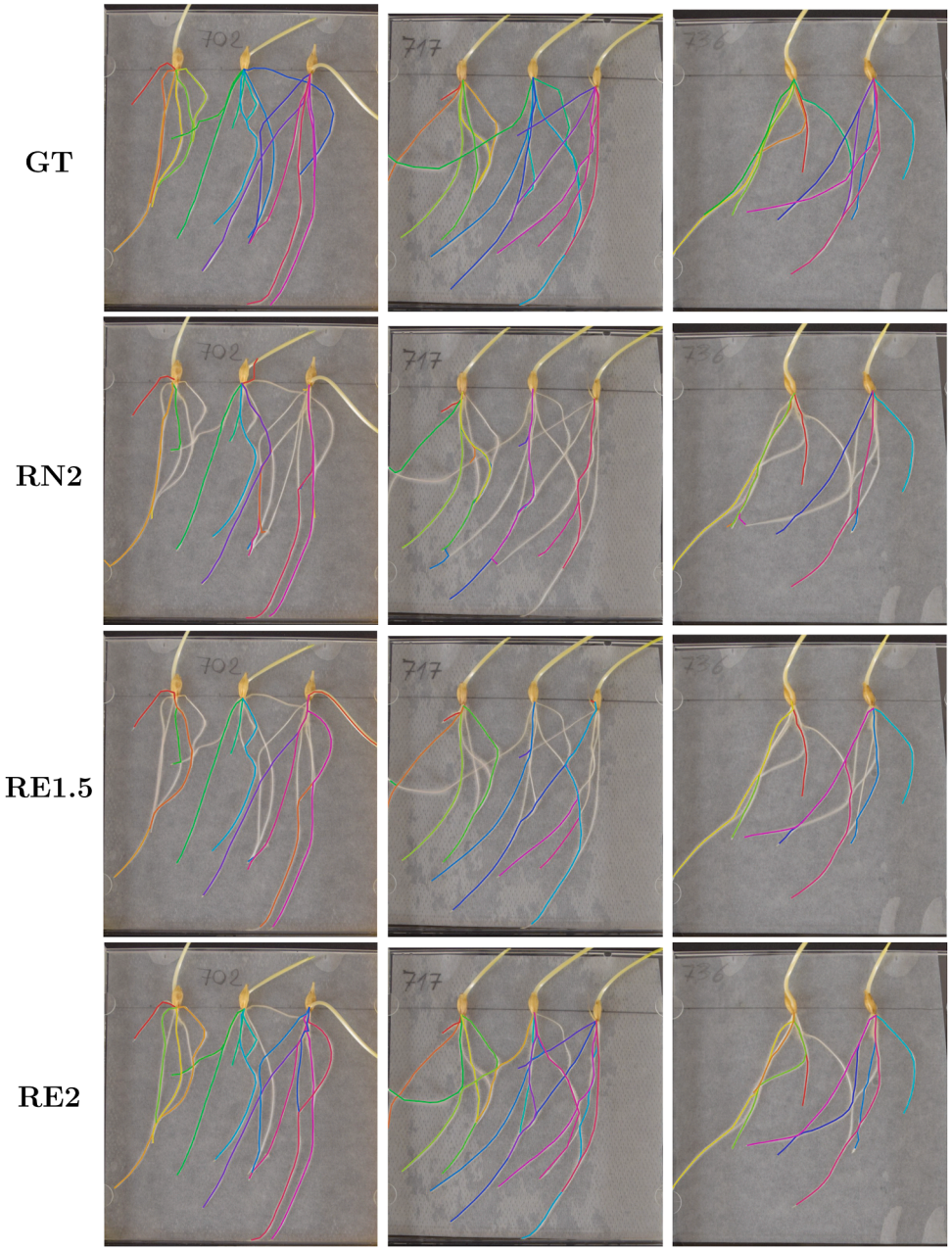


Fig. 12. Qualitative comparison for three samples of hard difficulty. Rows show the ground truth (GT) and predictions from **RN2**, **RE1.5** and **RE2**.

images is reported in Table 8. In the easy difficulty, tip-source connectivity is largely one-to-one and overlaps are rare, so all three methods reconstruct the gross architecture with relatively small errors. RN2 detects tips reliably, yet its unconstrained path-walking is prone to angular drift; because most tips are linked to their source by a single clear route, this weakness is partially masked but still inflates distance-based metrics. RE1.5 reverses the situation: its heuristic walking follows root trajectories more faithfully, but the absence of an explicit tip-localisation module causes it to miss tips that overlap thicker roots, again yielding moderate errors despite the simplicity of the scenes. RE2 combines both strengths, probabilistic tip detection and angle aware pathwalking and therefore reduces every discrepancy index, establishing a new state of the art for sparse root system.

In the medium difficulty, structural complexity rises appreciably: root segments overlap more frequently, several paths are truncated by the CD case border, and primary axes extend farther than in easy scenes. As shown in Fig. 11 and in Table 8, these factors accentuate the behavioural gap between methods.

- **RN2** still localises many tips, but its unconstrained pathwalking fails to negotiate dense intersections, producing angularly inconsistent trajectories and a high rate of missing or fragmented laterals.
- **RE1.5** follows individual trajectories more faithfully; however, the absence of an explicit tip detection stage prevents it from recovering tips embedded in thicker roots, yielding systematic under segmentation.
- **RE2** remains robust: the probabilistic tip detector retrieves most tips even in cluttered regions, while the angle aware walker limits drift across overlaps. A few path permutations relative to the ground truth persist especially where nearly parallel roots share entry points indicating that the current optimal path module can still misresolve geometrically similar alternatives.

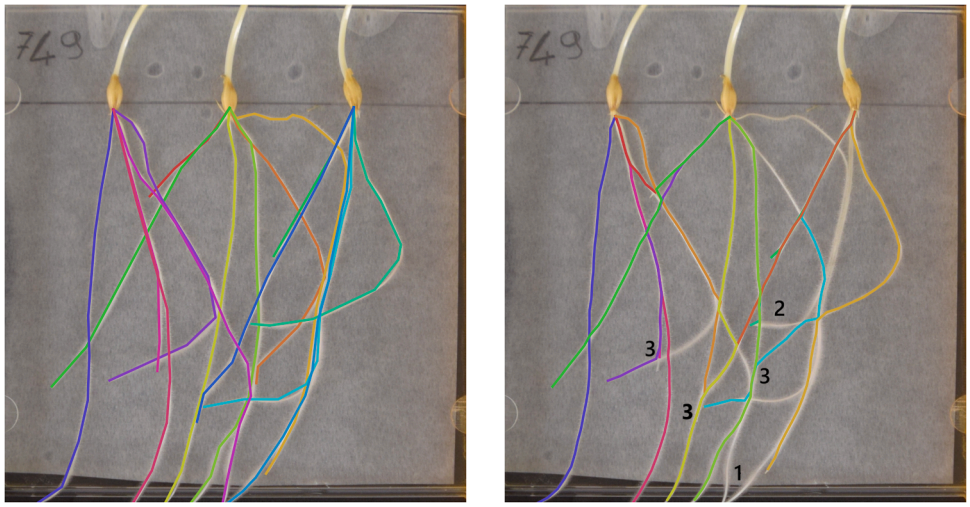


Fig. 13. Example of failure cases. On the left, we show the ground truth annotations; on the right, the corresponding output of the RN2 reconstruction pipeline. The colored lines represent individual root paths and are matched between ground truth and prediction during evaluation. Labels 1–3 in (b) highlight typical failure cases..

Overall, RE2 attains the lowest CA, TL, RT and OAD errors for this difficulty, yet the residual permutations highlight the need for a more globally consistent assignment strategy when root density and border truncations increase. In the hard difficulty the scene complexity reaches its peak: root segments overlap heavily, several tips are occluded or truncated at the cassette border, and the sheer number of individual roots amplifies topological ambiguity (Fig. 12).

- **RN2** produces only fragmentary graphs; its unconstrained walker cannot disambiguate tightly intertwined paths, leading to extensive omissions and the highest OAD and CA errors of all methods.
- **RE1.5** follows trajectories more accurately than RN2, but without a dedicated tip-detection module it misses numerous tips concealed within dense clusters, and its path estimates deviate when roots are abruptly truncated by the image frame.
- **RE2** handles the clutter best: tip detector retrieves most hidden tips, and the angle-aware walker maintains greater geometric consistency through intersections. Nevertheless, border-truncation still shortens some long axes and occasional path permutations remain, evidencing the residual challenge posed by extreme overlap.

Despite these limitations, RE2 secures four of the five best scores in the hard difficulty, demonstrating that the combined improvements in tip localisation and constrained path selection translate into substantial robustness under the most demanding imaging scenarios.

5.1. Failure cases and limitations

To better understand the limitations of our approach, we provide a qualitative analysis of typical failure cases, comparing the ground truth annotations with the results produced by our method.

Fig. 13 highlights three typical failure cases observed in our method, each labeled accordingly. The first case (labeled 1) illustrates a situation in which root tips are truncated by the image boundary, leading the algorithm to miss their detection. This type of failure could be mitigated by introducing a dedicated module capable of recognizing truncated roots, for instance through analysis of the skeleton structure near image edges. The second case (labeled 2) shows an error in the reconstruction of root trajectories caused by frequent physical contacts between roots. These contacts distort the expected growth paths and, in many instances, the tips are located close to intersections. This proximity increases ambiguity and leads the path-walking algorithm to select incorrect trajectories. The third case (labeled 3) highlights missing tips that are present in the ground truth but not detected by the algorithm. These omissions are mostly due to the small size of the tips or their overlap with nearby root segments, which makes them difficult to distinguish during the detection phase.

6. Conclusions

This work introduces an integrated framework for automated barley root phenotyping, delivering significant advances in both pixel-wise segmentation and graph-based structural reconstruction. The proposed *DeepRoot-3H* model, built on a multi-head DeepLabV3+ architecture, achieves state-of-the-art performance across various segmentation and keypoint detection metrics, outperforming both traditional pipelines and previous network versions.

By decoupling the learning objectives into dedicated heads for roots, tips, and sources, the architecture enhances both the delineation of fine structures and the localization of biologically relevant points—even in complex, overlapping regions. This is further reinforced by a training strategy that combines balanced loss terms and targeted class weighting, effectively addressing class imbalance and boosting detection robustness. The integration of post-processing refinements, such as contour-based filtering and area-guided tip separation, contributes to more complete and biologically coherent segmentations, ultimately improving the quality of RSML generation.

In parallel, the path extraction module leverages an iterative skeleton-based analysis coupled with a robust path-walking algorithm to reconstruct the topology of root systems. It incorporates biologically informed heuristics and geometric constraints to mitigate common issues such as over-segmentation, spurious branches, and inconsistent spacing. Its close synergy with accurate keypoint predictions ensures reliable anchoring and alignment of paths to relevant root tips and sources.

Together, these components form a unified system that not only advances the precision and interpretability of automated root phenotyping but also provides a scalable and generalizable tool for large-scale plant trait analysis. While the current implementation is tailored to controlled imaging conditions—such as hydroponic or gel-based media where root systems are clearly visible and unoccluded—we acknowledge the importance of addressing the additional challenges posed by roots imaged in soil, including fragmentation and occlusion. Future developments may incorporate broken-root repair algorithms or topology-aware inpainting techniques to expand the applicability of the framework to soil-based environments. Additional extensions could also explore the detection of further structural elements (e.g., branching points) or incorporating

temporal dynamics, broadening the framework's relevance for both basic plant research and applied agricultural contexts. Although this study focuses specifically on barley, the modular and adaptable nature of the proposed pipeline makes it suitable for extension to other crop species, provided appropriate training data and domain-specific adjustments are available.

Availability of Code

To foster reproducibility and facilitate further research, we provide the full implementation of our method, along with training scripts and pre-trained models. The code is available at the following repository: https://github.com/MaicholD95/RootEx2.0

CRediT authorship contribution statement

Maichol Dadi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft; **Annalisa Franco:** Conceptualization, Data curation, Writing – review & editing, Visualization; **Alessandra Lumini:** Conceptualization, Data curation, Methodology, Validation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Data availability

Data and code are shared on github.

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.

Funding

This study was carried out within the Agritech National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR)—MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4—D.D. 1032 17/06/2022, CN00000022). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

Bhugra, S., Mukherjee, P., Kaushik, V., Jha, R., Lall, B., & Chaudhury, S. (2022). Tarsnet: Topology aware root segmentation network for plant phenotyping. In *Proceedings of the thirteenth Indian conference on computer vision, graphics and image processing* (pp. 1–8).

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information*, *11*(2). https://doi.org/10.3390/info11020125

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018a). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the european conference on computer vision (ECCV)*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 801–818).

Dadi, M., Franco, A., Sangiorgi, G., Salvi, S., & Lumini, A. (2025). Tillmore cdc dataset. https://github.com/MaicholD95/TILLMore-CDC.

Dadi, M., Franco, A., Sangiorgi, G., Salvi, S., & Lumini, A. (2024a). Roottracer: An intuitive solution for root image annotation. *Smart Agricultural Technology*, *10*, 100705. https://doi.org/10.1016/j.atech.2024.100705

Dadi, M., Lumini, A., & Franco, A. (2025). Rootex: An automated method for barley root system extraction and evaluation. *Computers and Electronics in Agriculture*, *230*, 110030. https://doi.org/10.1016/j.compag.2025.110030

Dadi, M., Lumini, A., Franco, A., & Sangiorgi, G. (2024b). Deep learning for root system extraction from barley plants. In *Proceedings of the IEEE 14th international conference on pattern recognition systems (ICPRS)*. London, UK.

Gaggion, N., Ariel, F., Daric, V., Lambert, E., Legendre, S., Roule, T., Camoirano, A., Milone, D. H., Crespi, M., Blein, T., & Ferrante, E. (2021). Chronoroot: High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. *GigaScience*, *10*(7), giab052. https://doi.org/10.1093/gigascience/giab052

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).

Judd, L. A., Jackson, B. E., & Fonteno, W. C. (2015). Advancements in root growth measurement technologies and observation capabilities for container-grown plants. *Plants*, *4*(3), 369–392.

Leitner, D. et al. (2012). Smartroot: An open source root image analysis tool. *Plant physiology*, *160*(1), 487–498.

Li, H., Qi, J., Tang, S. et al. (2021). Hsnet: Hierarchical pyramid scene parsing network for accurate and efficient segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 7455–7464).

Liu, G., Yao, S., Liu, D., Chang, B., Chen, Z., Wang, J., & Wei, J. (2024). Cafe-net: Cross-attention and feature exploration network for polyp segmentation. *Expert Systems with Applications*, *238*, 121754. https://doi.org/10.1016/j.eswa.2023.121754

Lobet, G., Pound, M. P., Diener, J., Pradal, C., Draye, X., Godin, C., Javaux, M., Leitner, D., Meunier, F., Nacry, P., Pridmore, T. P., & Schnepf, A. (2015). Root system markup language: Toward a unified root architecture description language. *Plant Physiology*, *167*(3), 617–627.

Mingxuan, Z., Wei, L., Hui, L., Ruinan, Z., & Yiming, D. (2022). Anti-gravity stem-seeking restoration algorithm for maize seed root image phenotype detection. *Computers and Electronics in Agriculture*, *202*, 107337. https://doi.org/10.1016/j.compag.2022.107337

Narisetti, N., Henke, M., Seiler, C., Junker, A., Ostermann, J., Altmann, T., & Gladilin, E. (2021). Fully-automated root image analysis (faRIA). *Scientific Reports*, *11*(1), 1–15.

Pagès, L., Picon-Cochard, C., & Bengough, A. G. (2014). A generic modelling framework for analysing and simulating the development of root system architecture. *Plant and Soil*, *372*(1), 93–111.

Pound, M. P., French, A. P., Atkinson, J. A., Wells, D. M., Bennett, M. J., & Pridmore, T. (2013). Rootnav: Navigating images of complex root architectures. *Plant Physiology*, *162*(4), 1802–1814.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, *9351*, 234–241.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, *9*(7), 676–682. https://doi.org/10.1038/nmeth.2019

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). Nih image to imagej: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675.

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), plab056.

Seidenthal, K., Panjvani, K., Chandnani, R., Kochian, L. V., & Eramian, M. G. (2022). Iterative image segmentation of plant roots for high-throughput phenotyping. *Scientific Reports*, *12*. https://api.semanticscholar.org/CorpusID:252685741.

Su, H., Liu, S., Zheng, B., Zhou, X., & Zheng, K. (2020). A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, *29*, 3–32.

Wang, C., Li, X., Caragea, D., Bheemanahallia, R., & Jagadish, S. V. K. (2020). Root anatomy based on root cross-section image analysis with deep learning. *Computers and Electronics in Agriculture*, *175*, 105549.

Wang, G., Zhang, L., Lu, P. et al. (2023). Cafe: Content-aware feature enhancement for accurate polyp segmentation. *IEEE Transactions on Medical Imaging*, *42*(1), 66–78.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 568–578).

Yasrab, R., Atkinson, J. A., Wells, D. M., French, A. P., Pridmore, T. P., & Pound, M. P. (2019). Rootnav 2.0: Deep learning for automatic navigation of complex plant root architectures. *GigaScience*, *8*(11), giz123.

Yu, Q., Wang, J., Tang, H., Zhang, J., Zhang, W., Liu, L., & Wang, N. (2023). Application of improved UNet and EnglightenGAN for segmentation and reconstruction of in situ roots. *Plant Phenomics*, *5*, 0066.