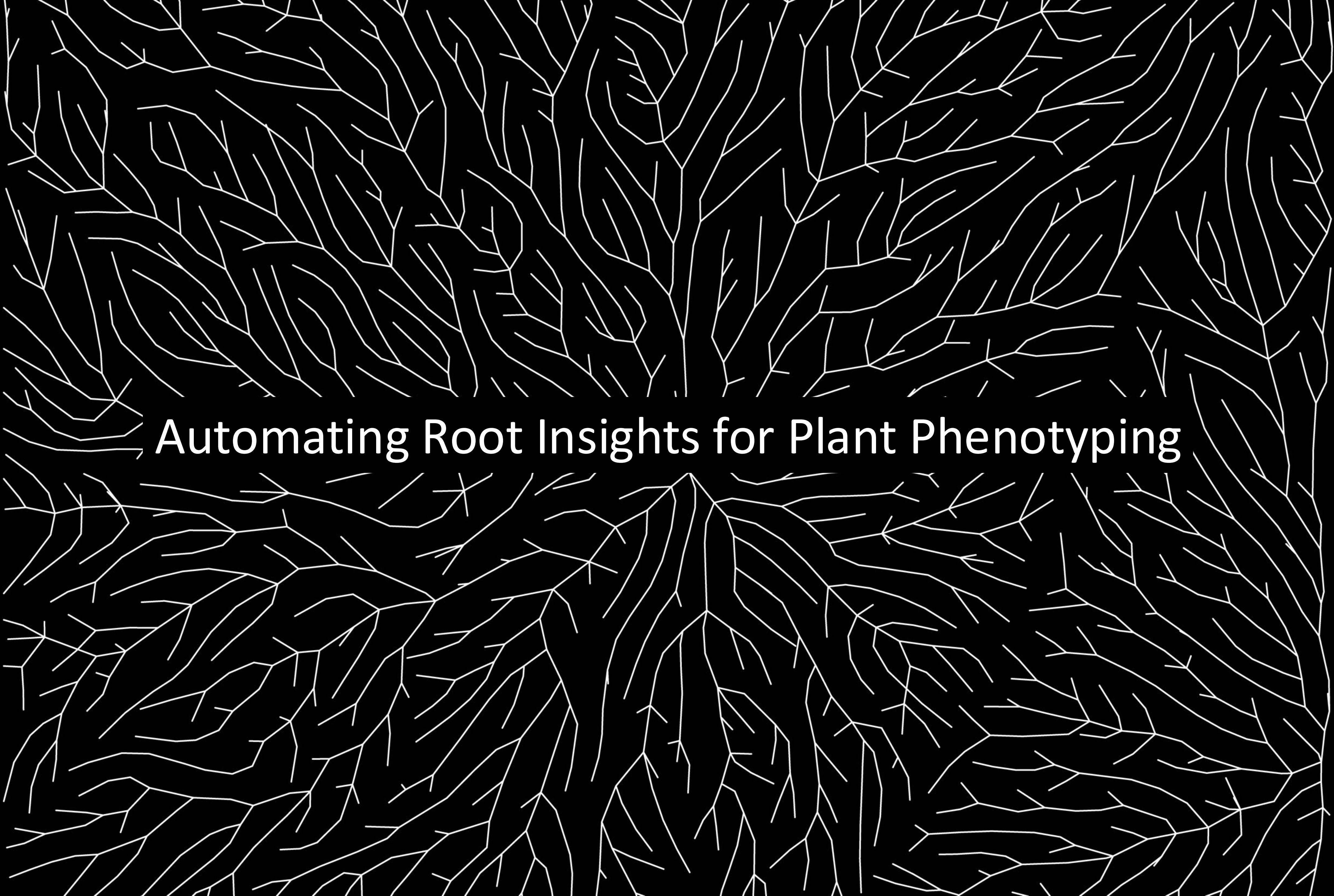
Breda University of Applied Sciences

Data Science & AI

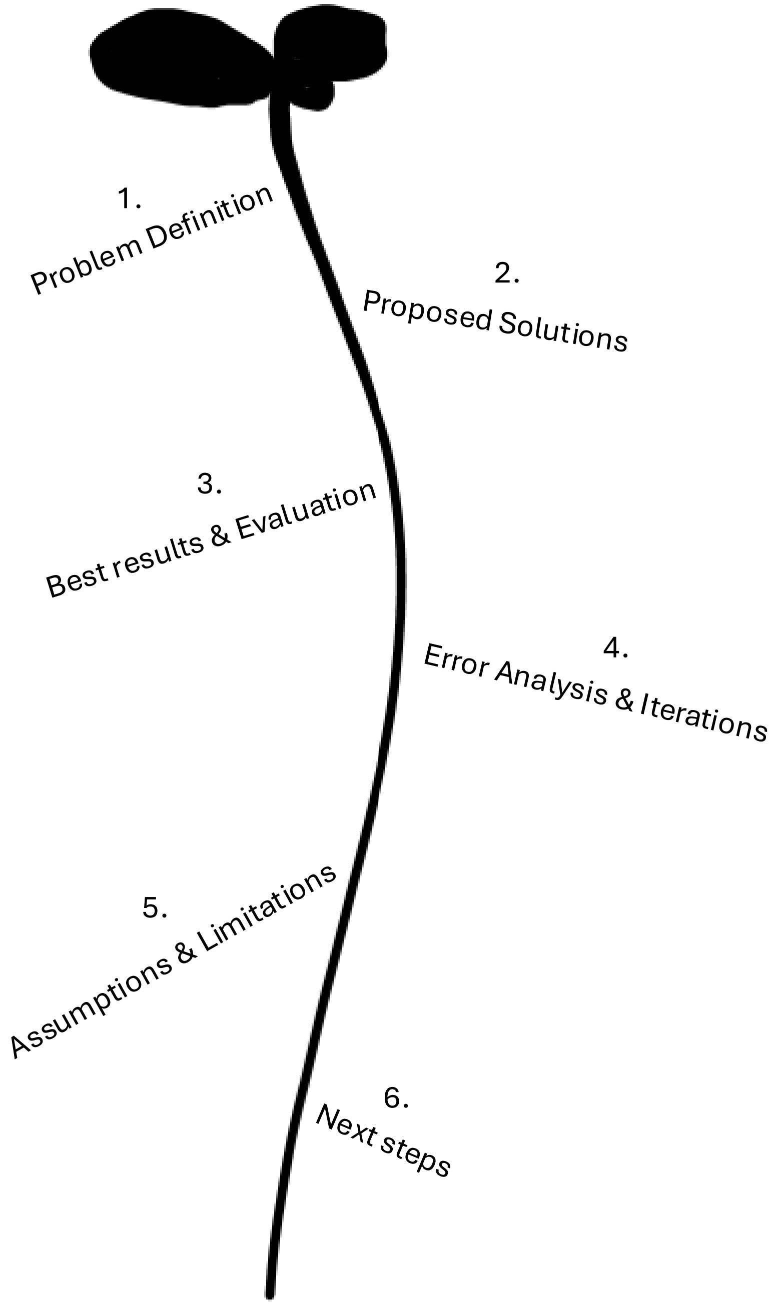
In collab with NPEC



Automating Root Insights for Plant Phenotyping

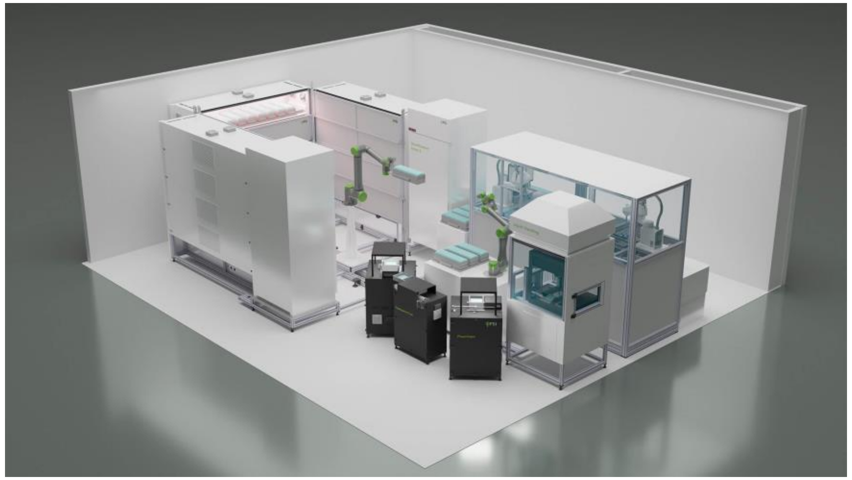
Louie Daans

Image by Garry Killian

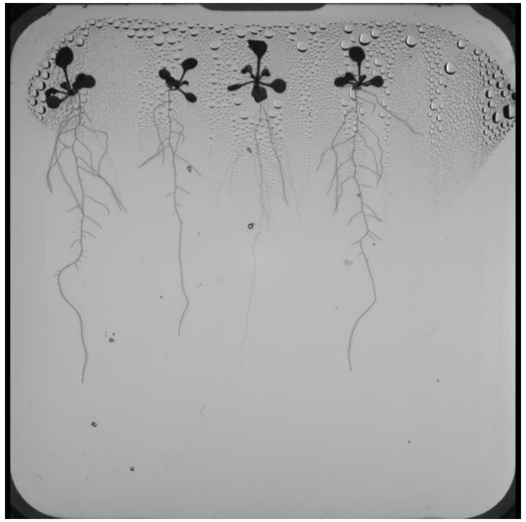

1.
Problem Definition
2.
Proposed Solutions
3.
Best results & Evaluation
4.
Error Analysis & Iterations
5.
Assumptions & Limitations
6.
Next steps

Problem Definition

The Netherlands Plant Eco-phenotyping Centre (NPEC) aims to advance plant research trough automated phenotyping systems.



Their Hades system is designed to automate imaging and analysis of in-vitro root systems. It captures daily images, enabling phenotyping of roots under controlled conditions. These images exhibit overlapping root structures, varying contrast, and noise, which make manual analysis labour-intensive and error-prone.



Data science offers solutions to automate and improve the segmentation process. By leveraging machine learning algorithms, we can develop models capable of accurately distinguishing roots, even in complex scenarios. This enables plant scientist to rapidly extract key phenotypic traits such as root length, branching and root tip coordinates, which can be integrated into a data pipeline to control a liquid-handling robot for precise inoculation.

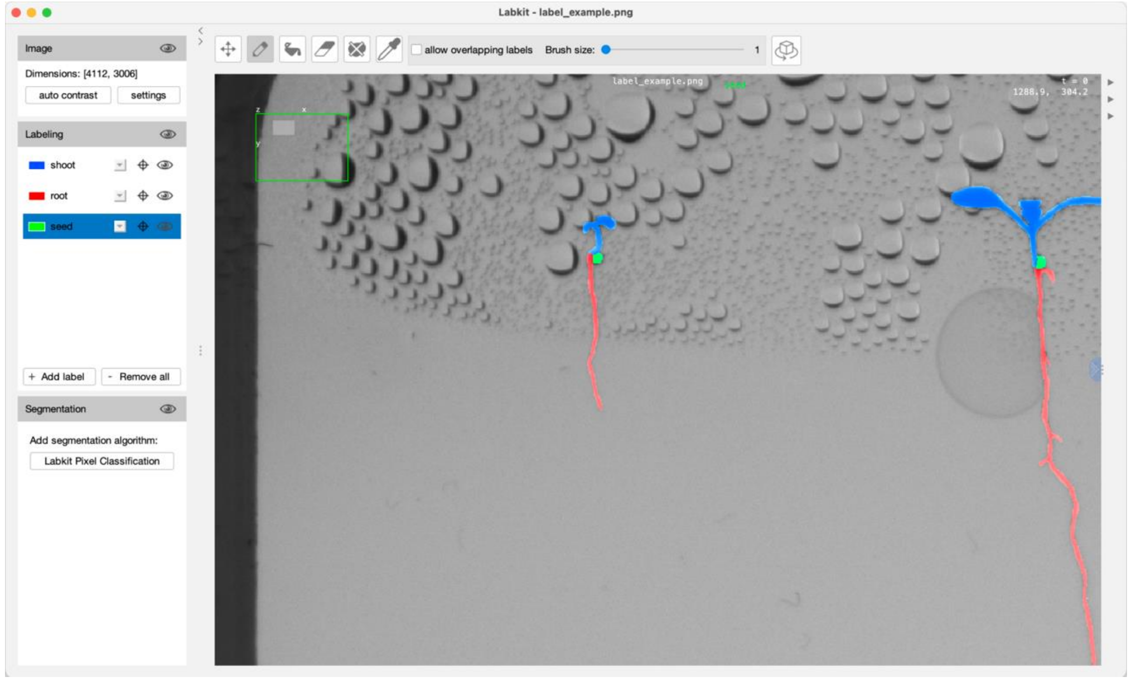
Root systems are fundamental for plant development. They anchor the plant, absorb nutrients and water, and interact with microbes in the soil. Studying root architecture, growth patterns, and interactions with beneficial or harmful microbes provides critical insights into how plants perform under various environmental conditions.

Proposed Solution

To attack this challenge, we need to automate the imaging, analysis, and interaction with plant root systems using a combination of data science, computer vision, and robotics.

1. Data Annotation:

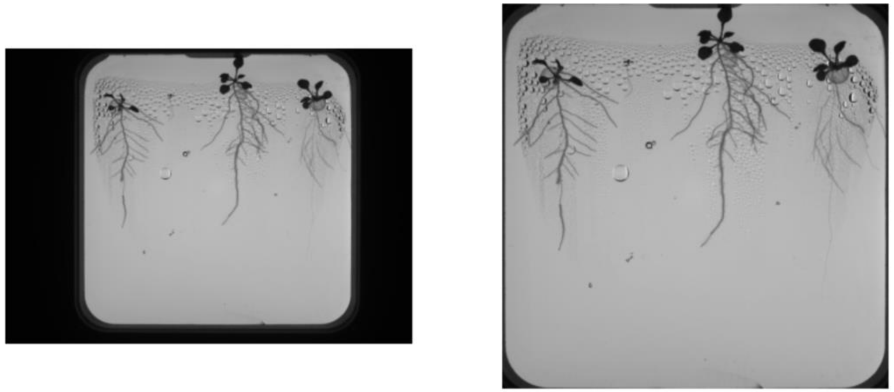
Collaborating between students to define the ‘truth’ by manually annotating shoots, roots and seeds within the images



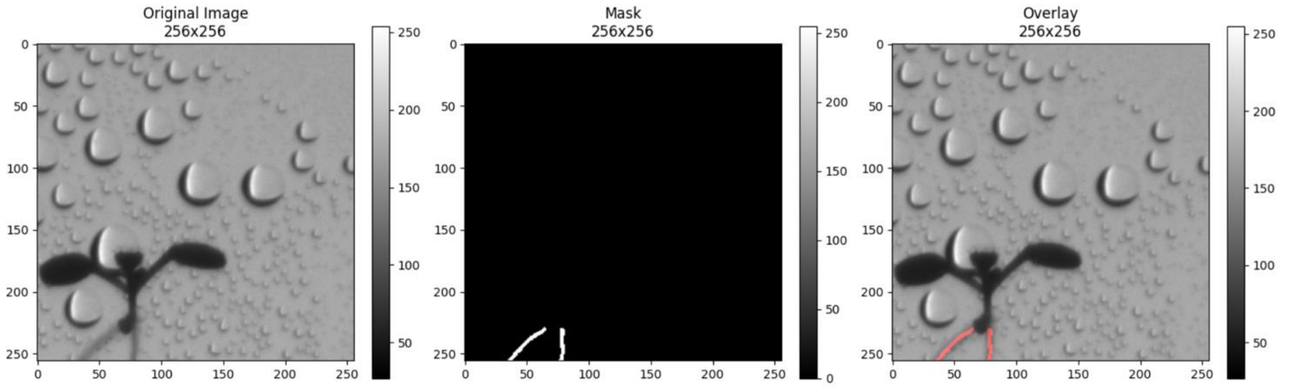
2. Data Preprocessing:

Process raw images and annotated masks to prepare data for machine learning (noise reduction, image resizing, patching, …)

(2736, 2736)



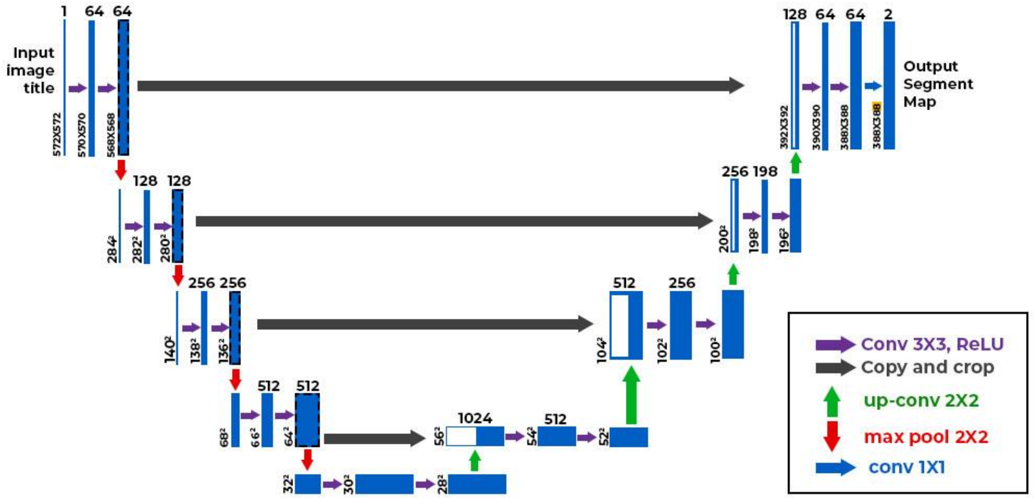
cropping

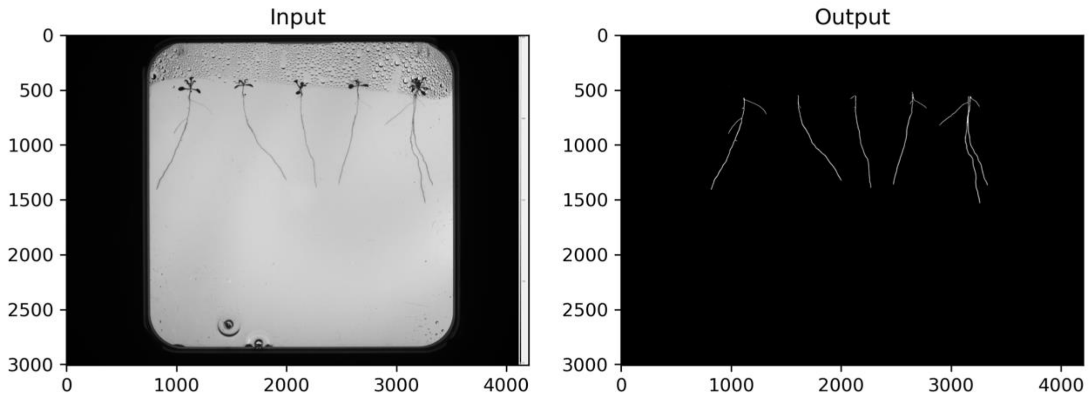


256x256 patching

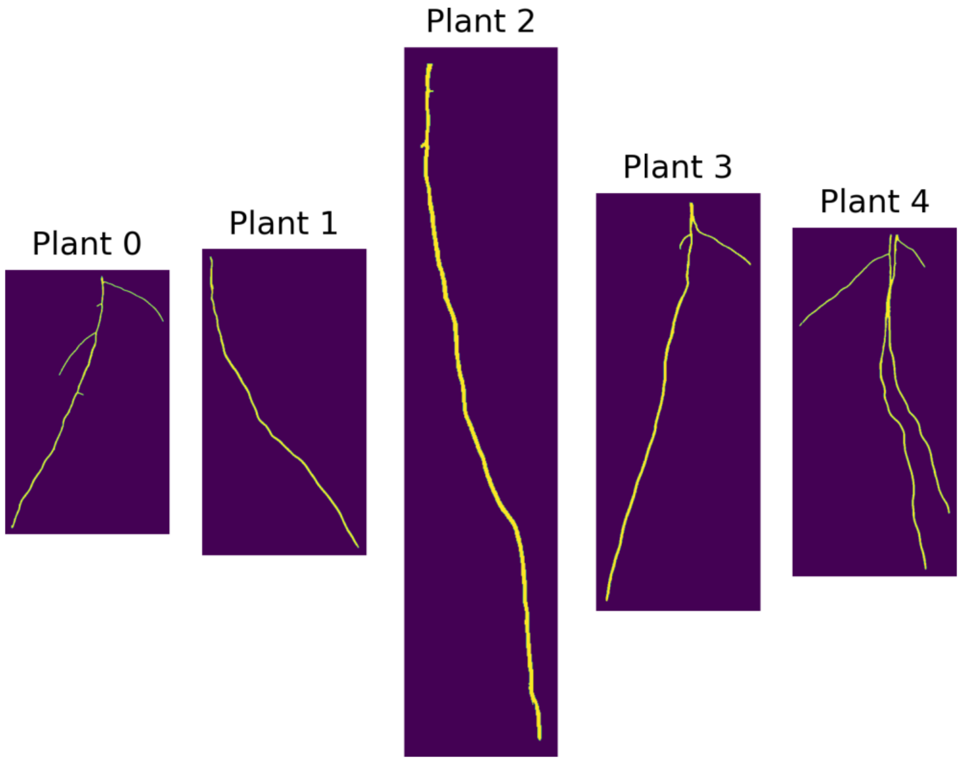
Proposed Solution

3. Model Training:
Use U-NET deep learning architecture for root segmentation and tune hyperparameters until satisfied.



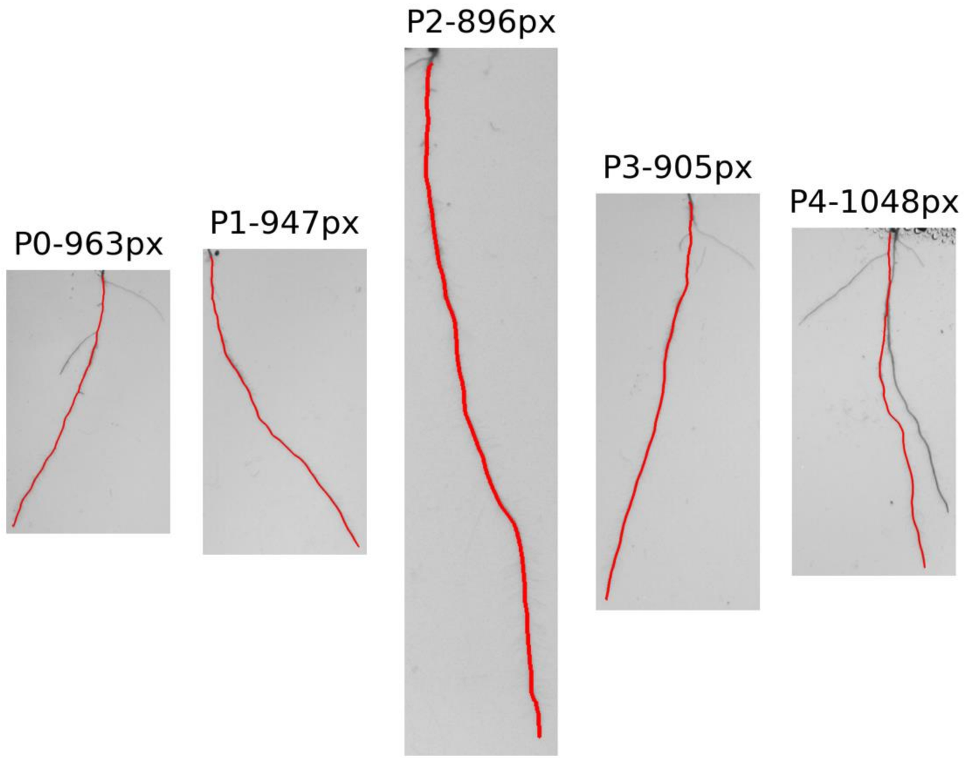


4. Root Segmentation:
Using traditional CV methods to assign root systems to individual plants



Proposed Solution

5. Root Metrics Extraction:
Extract key root traits for each plant; root length and main root tip coordinates.

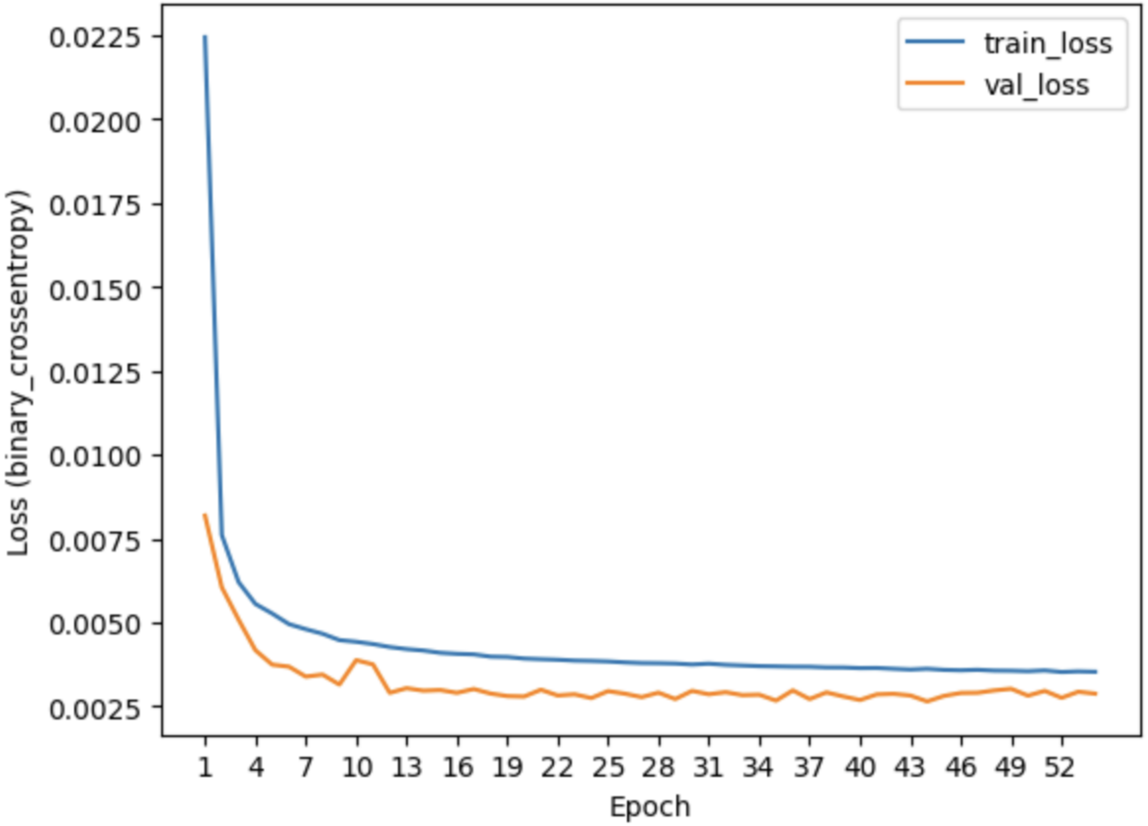


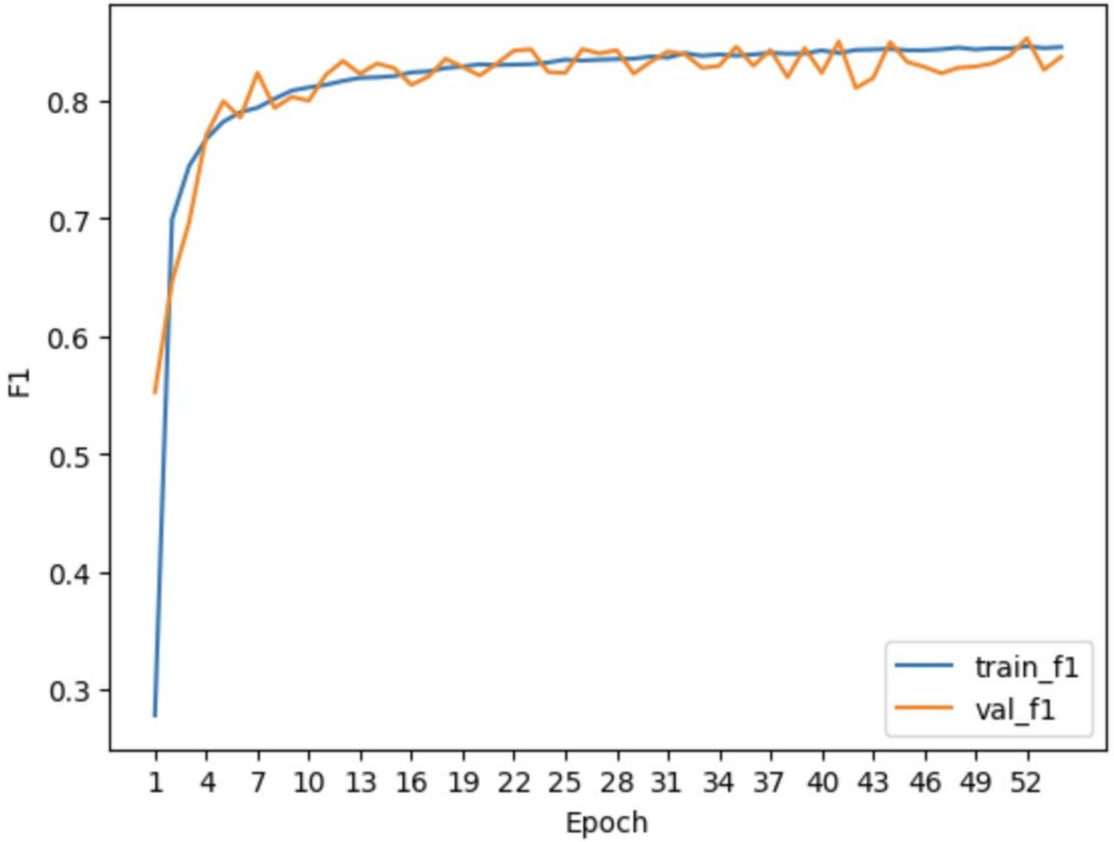
6. Robotics Integration:
Train a robot to move precisely to the identified root tip locations and automate the dispensing of liquid.

Best results & Evaluation

Model Training

Root Validation F1: 0.85



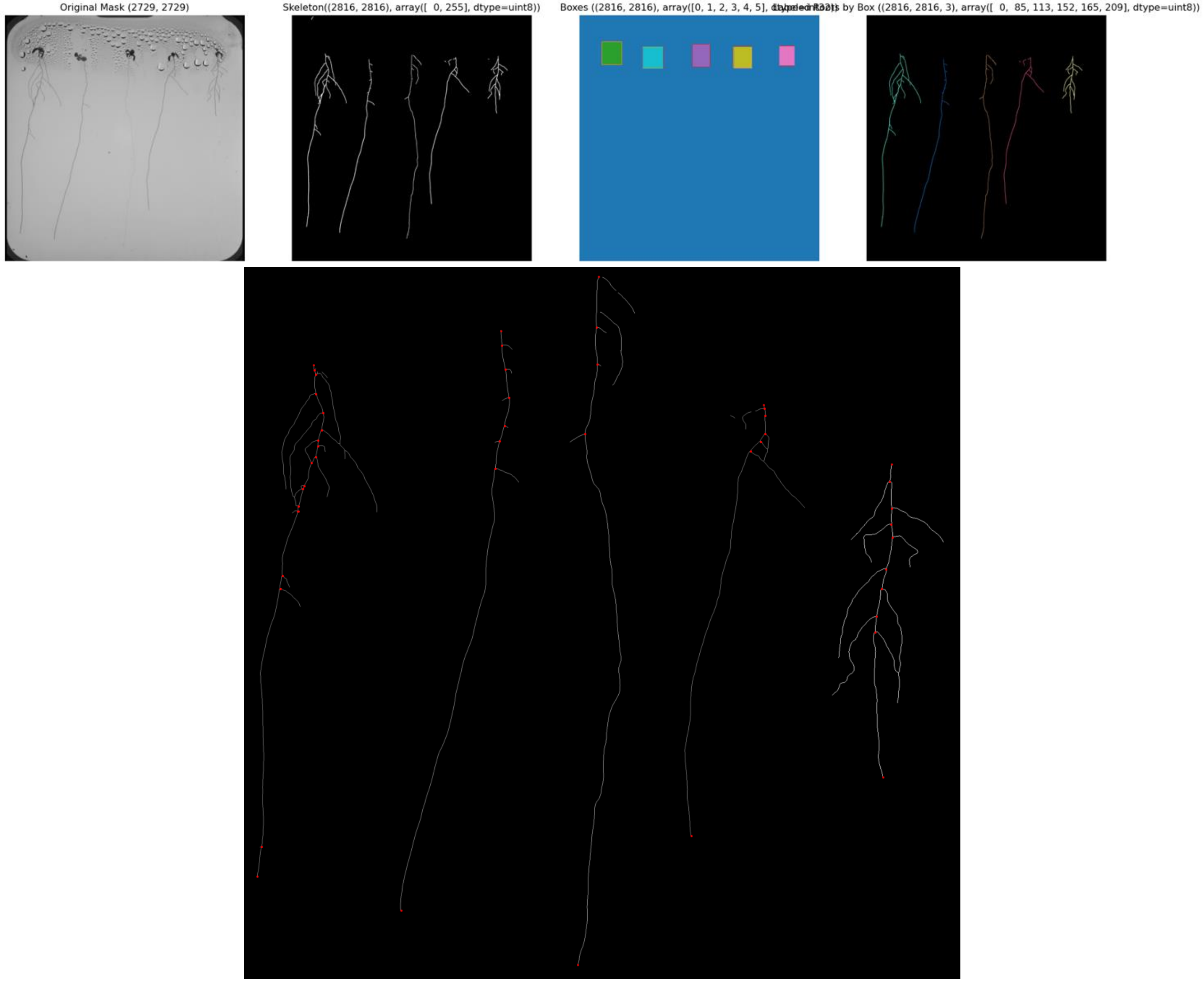


Best results & Evaluation

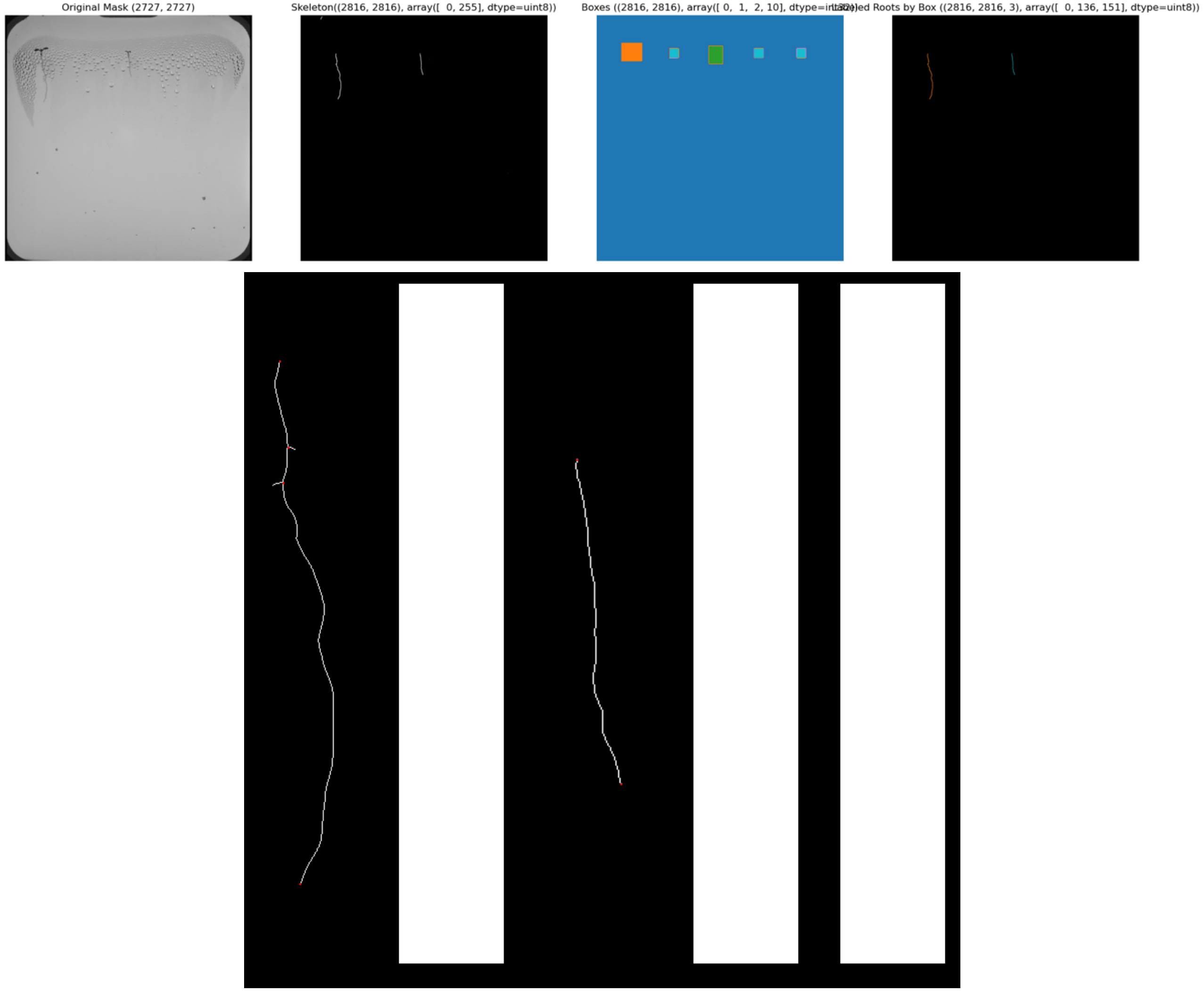
Primary root length predictions

Symmetric Mean Absolute Percentage Error (SMAPE) of 6.37

Example 1



Example 2



Best results & Evaluation

Robotics

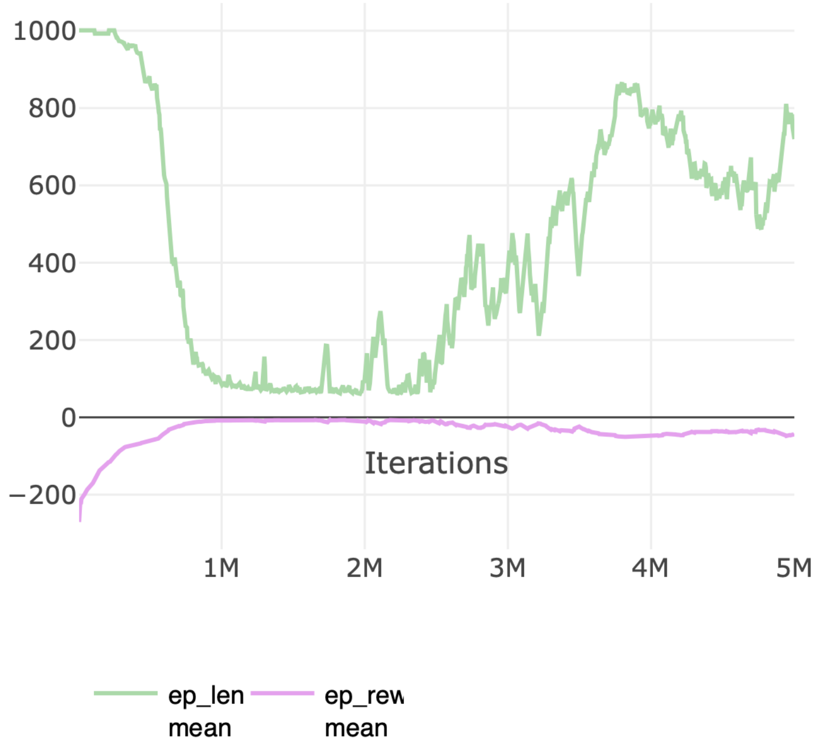
RL controller
(1mm threshold)

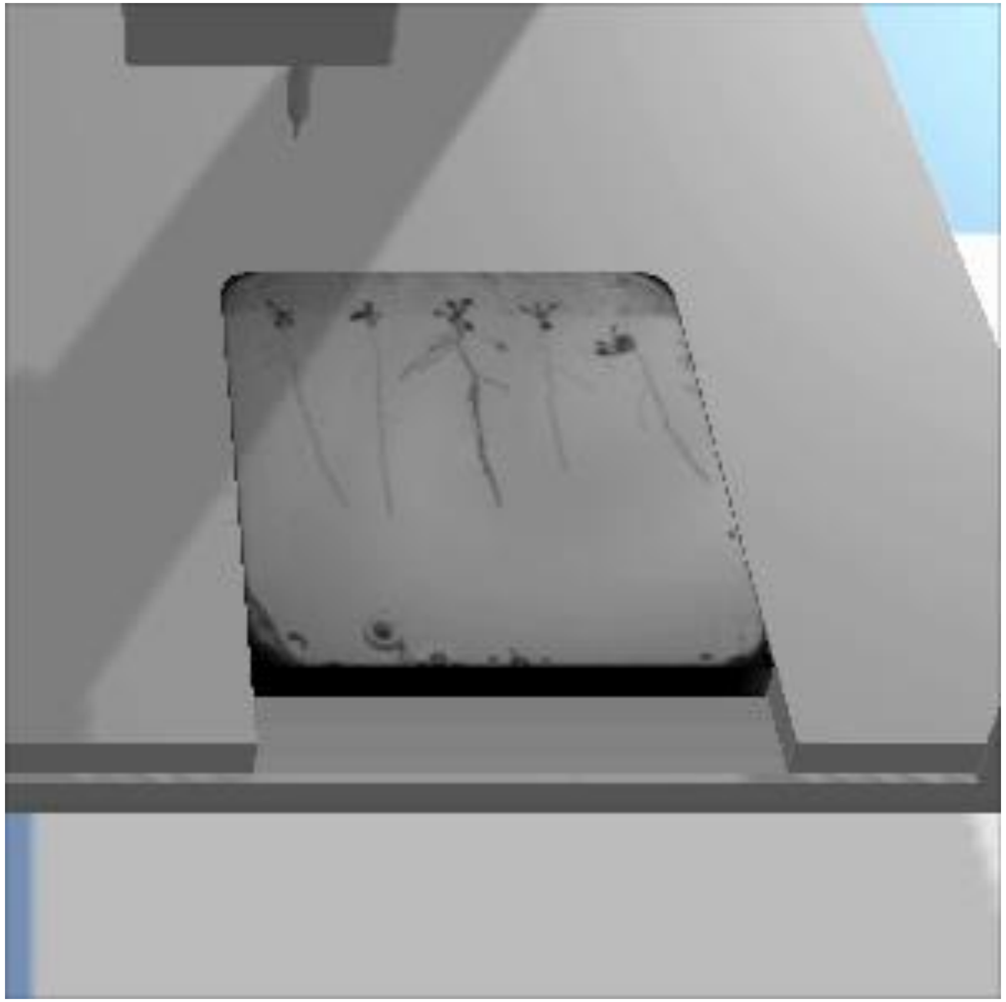
[Mean Episode Length -> 62]

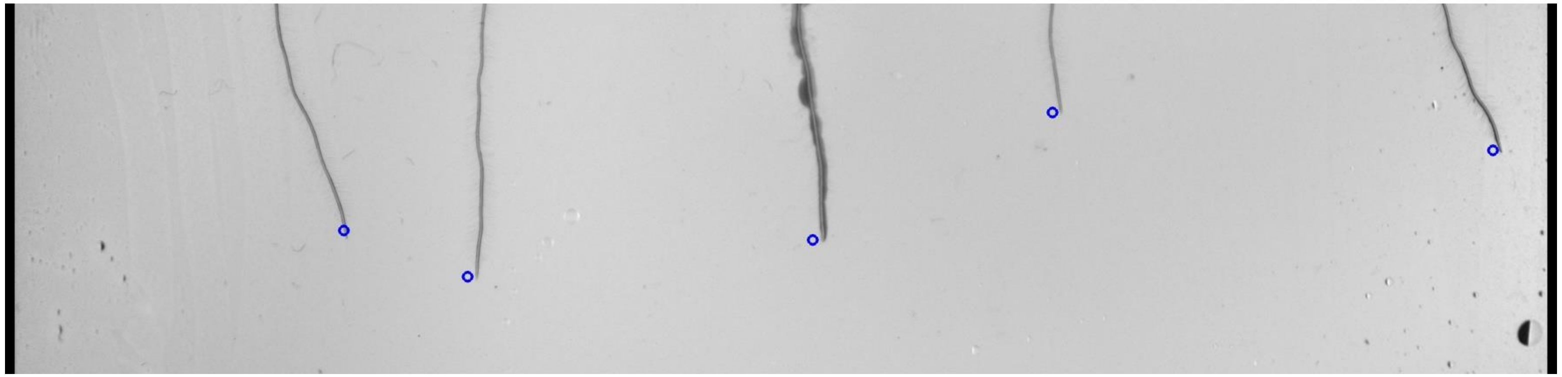
100% success on 18 test petri dishes

Rewards:

- Punish negative distance to goal
- Punish every step
- Punish not improving distance to goal
- Reward success





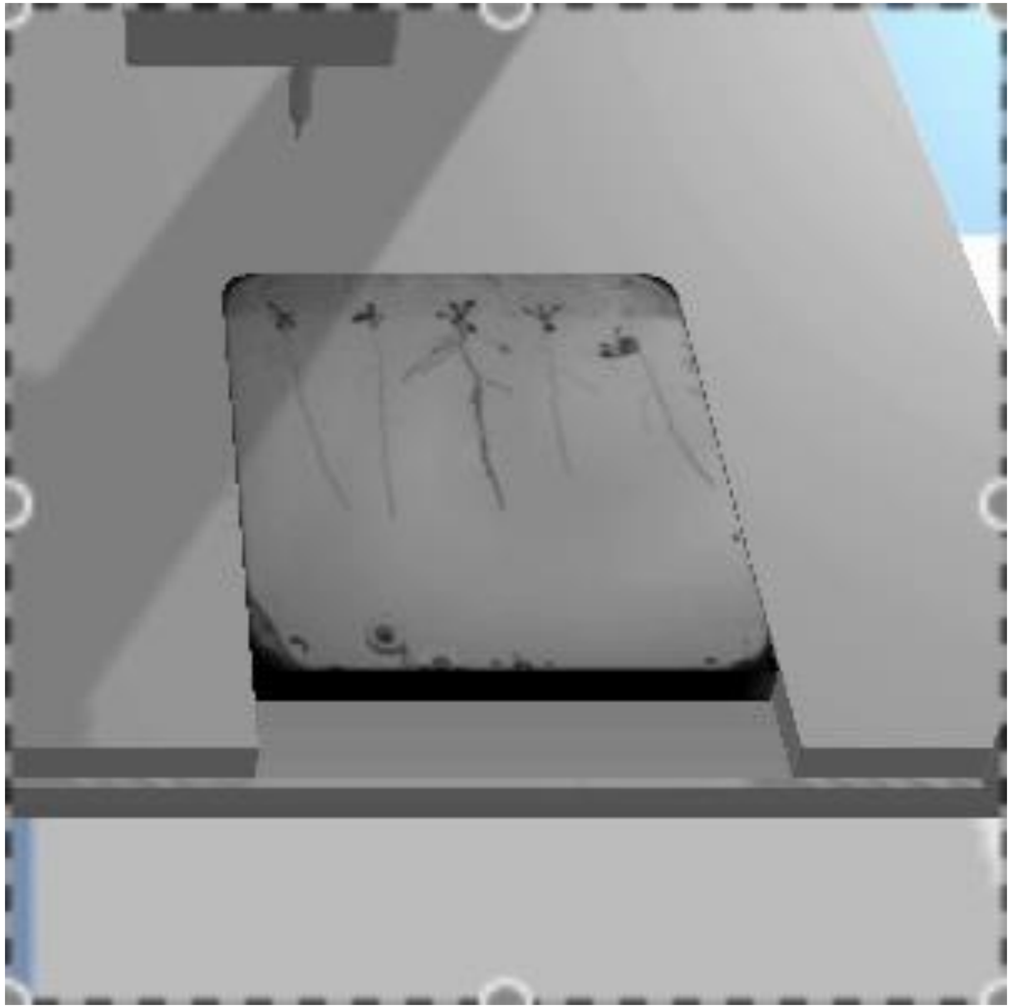


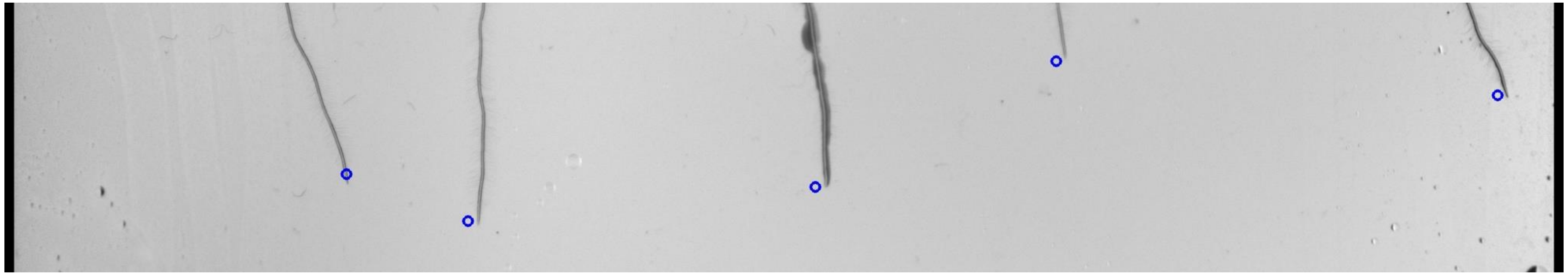
Best results & Evaluation

Robotics

PID controller
(1mm threshold)

100% success on 18 test petri dishes





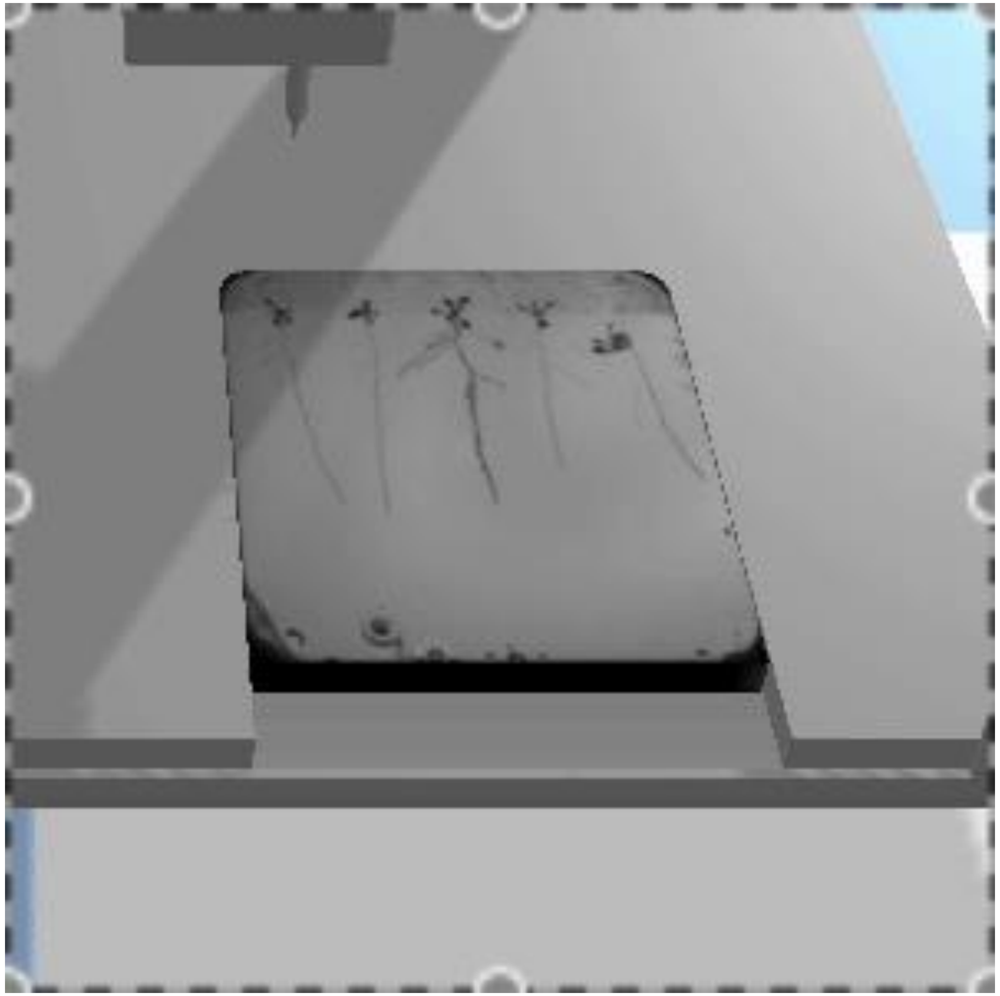
Best results & Evaluation

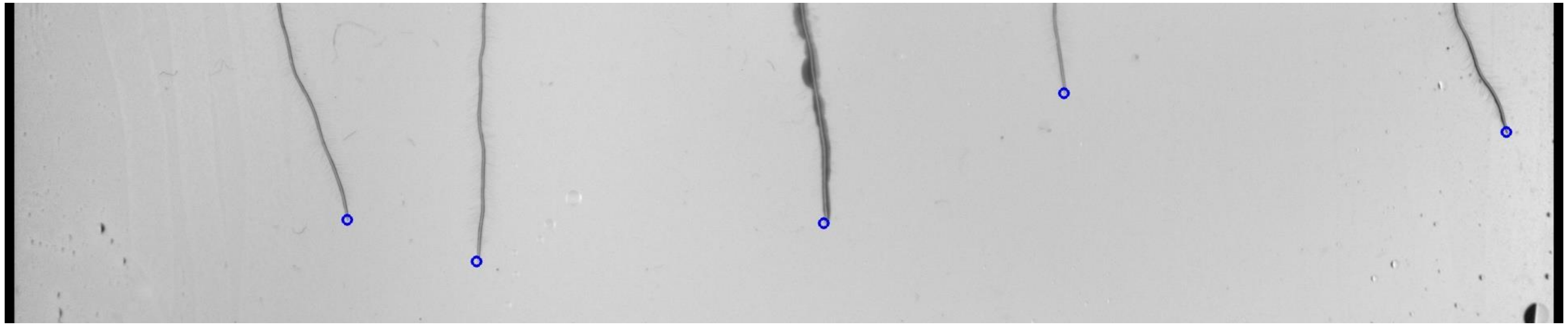
Robotics

PID controller
(0.1mm threshold)

(best performing controller)

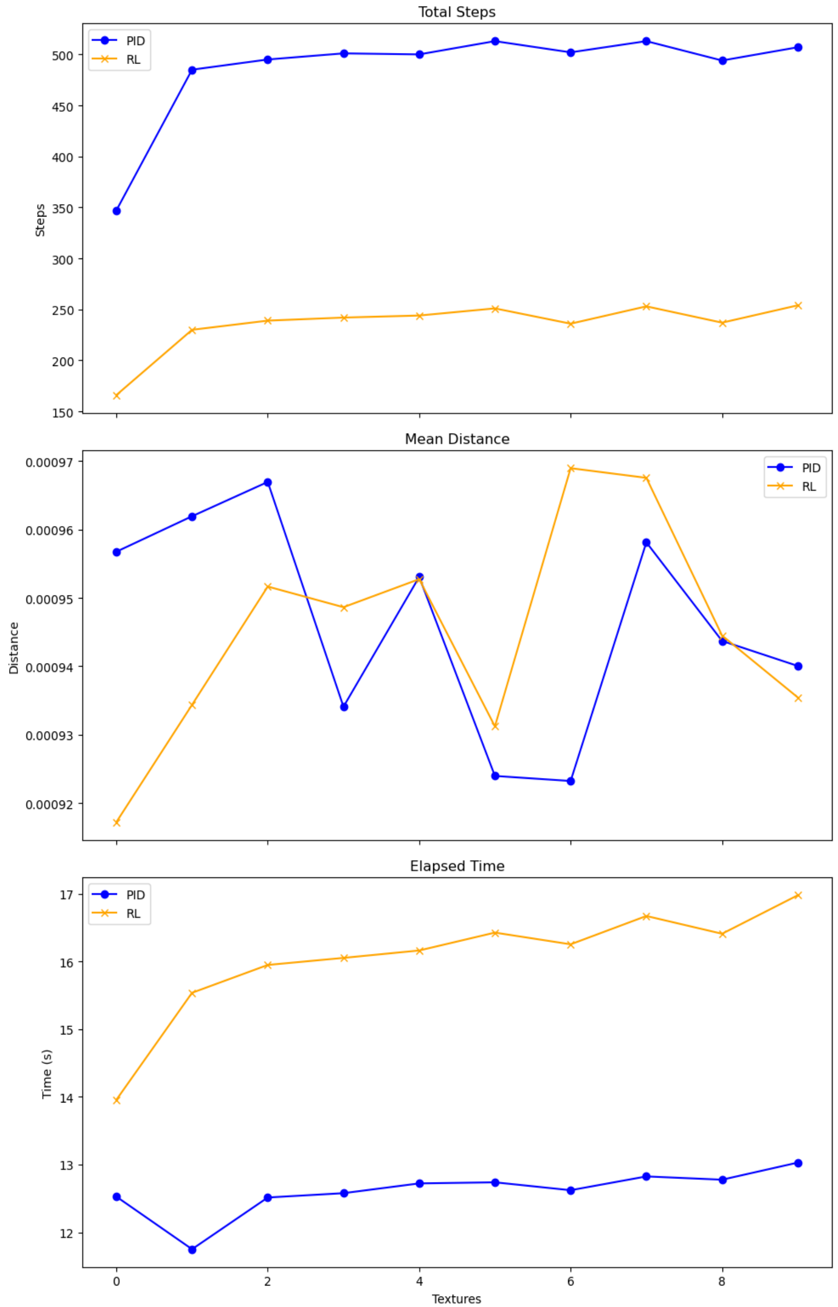
100% success on 18 test petri dishes

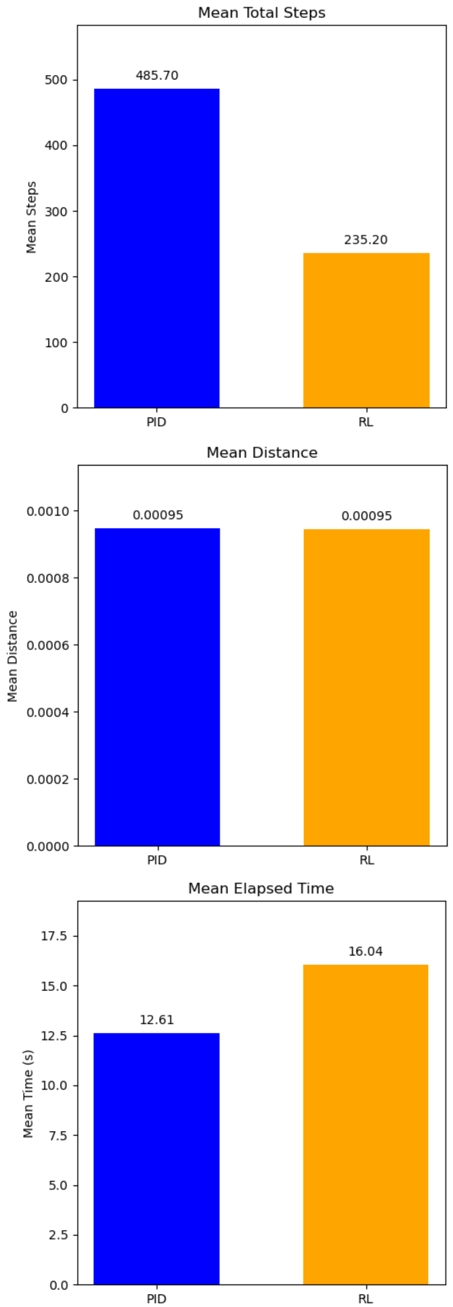


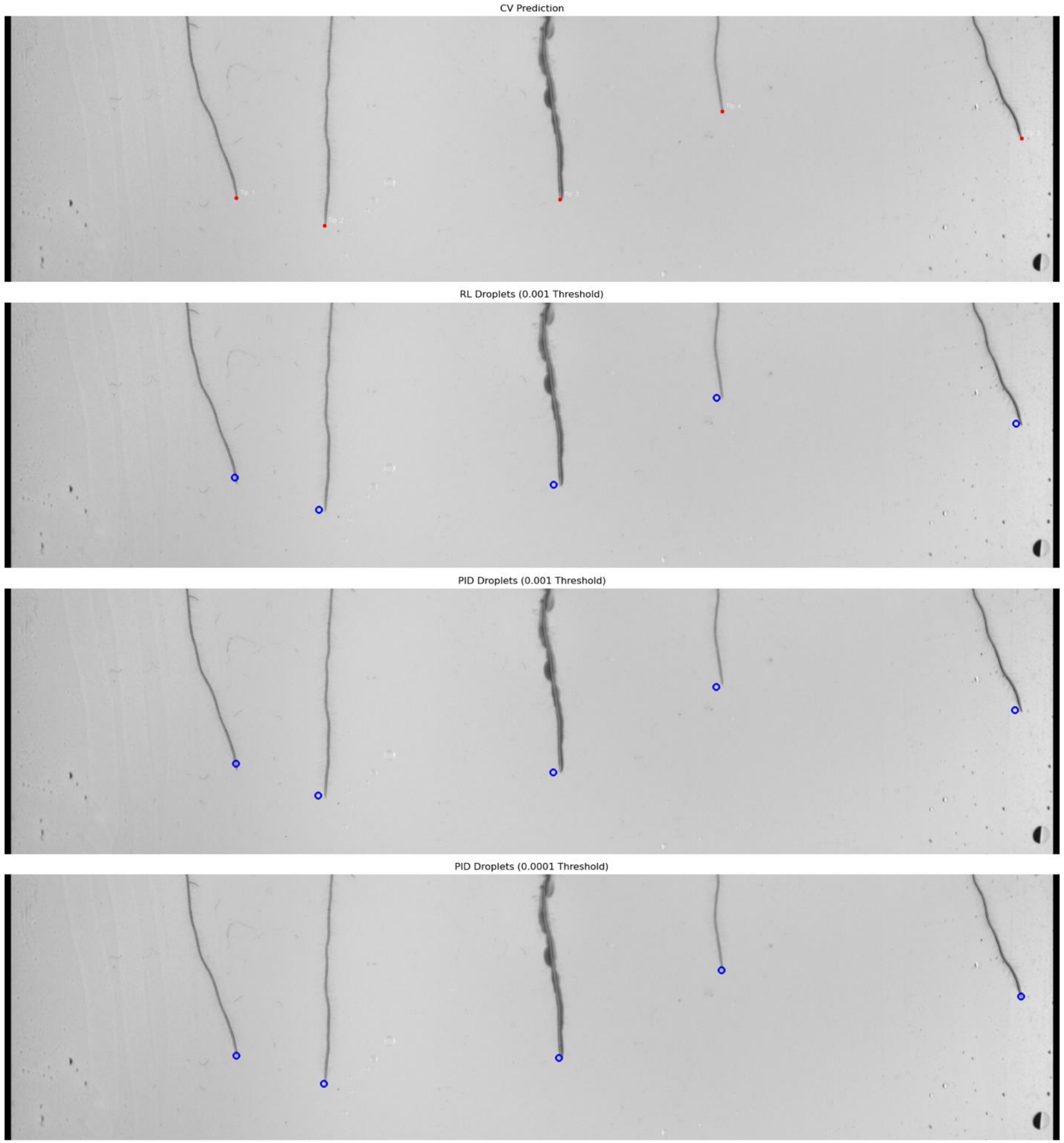


Best results & Evaluation

Comparing RL with PID







Comparing RL with PID

Although the performances of the two controllers appear similar based on metrics, rendering the simulation reveals significant differences in their behaviour. The RL controller exhibits unsmooth movements, which could potentially cause imprecise landings of the droplets on the petri dish. In contrast, the PID controller demonstrates smoother movements, particularly as it approaches the target. This allows for greater precision when placing the droplet.

While the PID controller requires more total steps to complete the task, this is an advantage in this context, as it prioritizes accuracy and control.

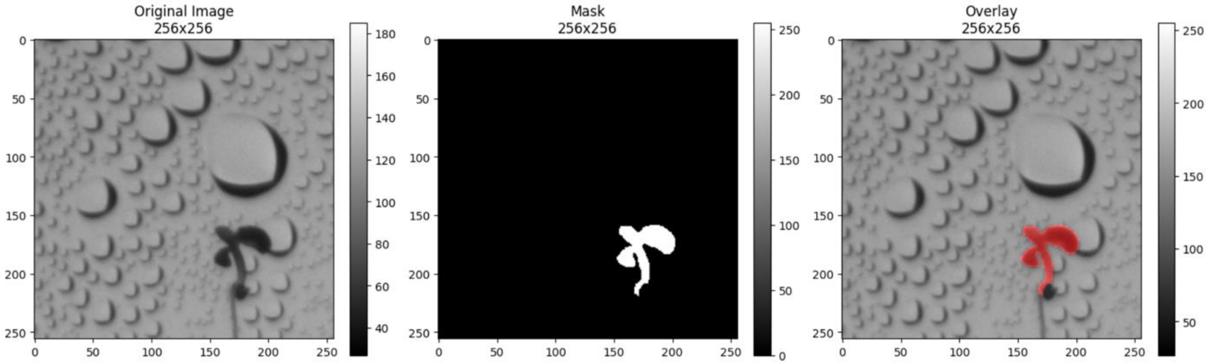
An additional advantage of the PID controller is the ease with which its precision can be adjusted. For example, reducing the threshold from 1mm to 0.1mm can be achieved with minimal impact on time or the number of steps required. Achieving the same level of precision with an RL model, however, would require significantly more computational effort and training time.

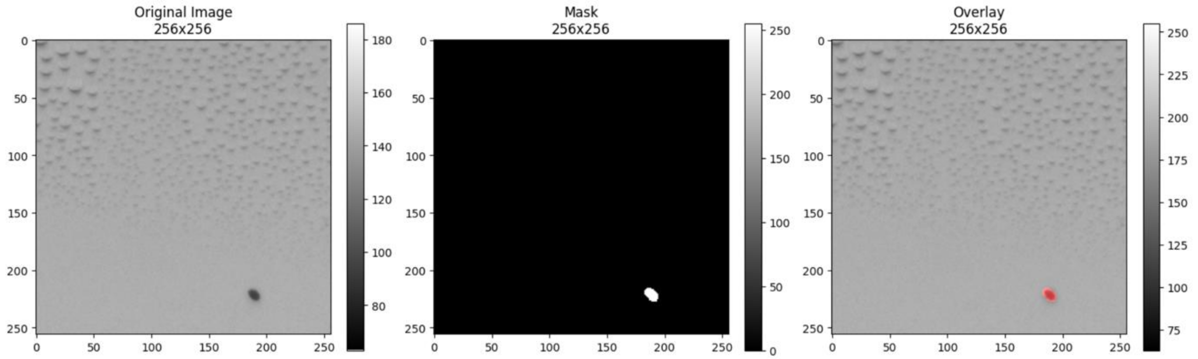
In conclusion, the PID controller proves to be highly effective for this task. Given its simplicity, reliability, and performance, investing significant resources and time into RL for straightforward tasks like these may not be necessary.

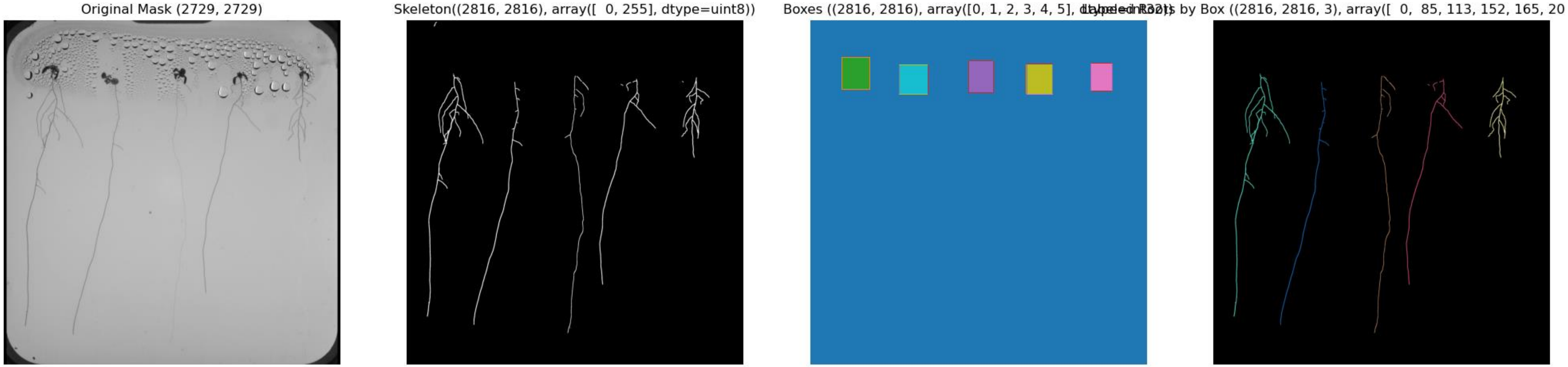
Error Analysis & Iterations

Iteration 1

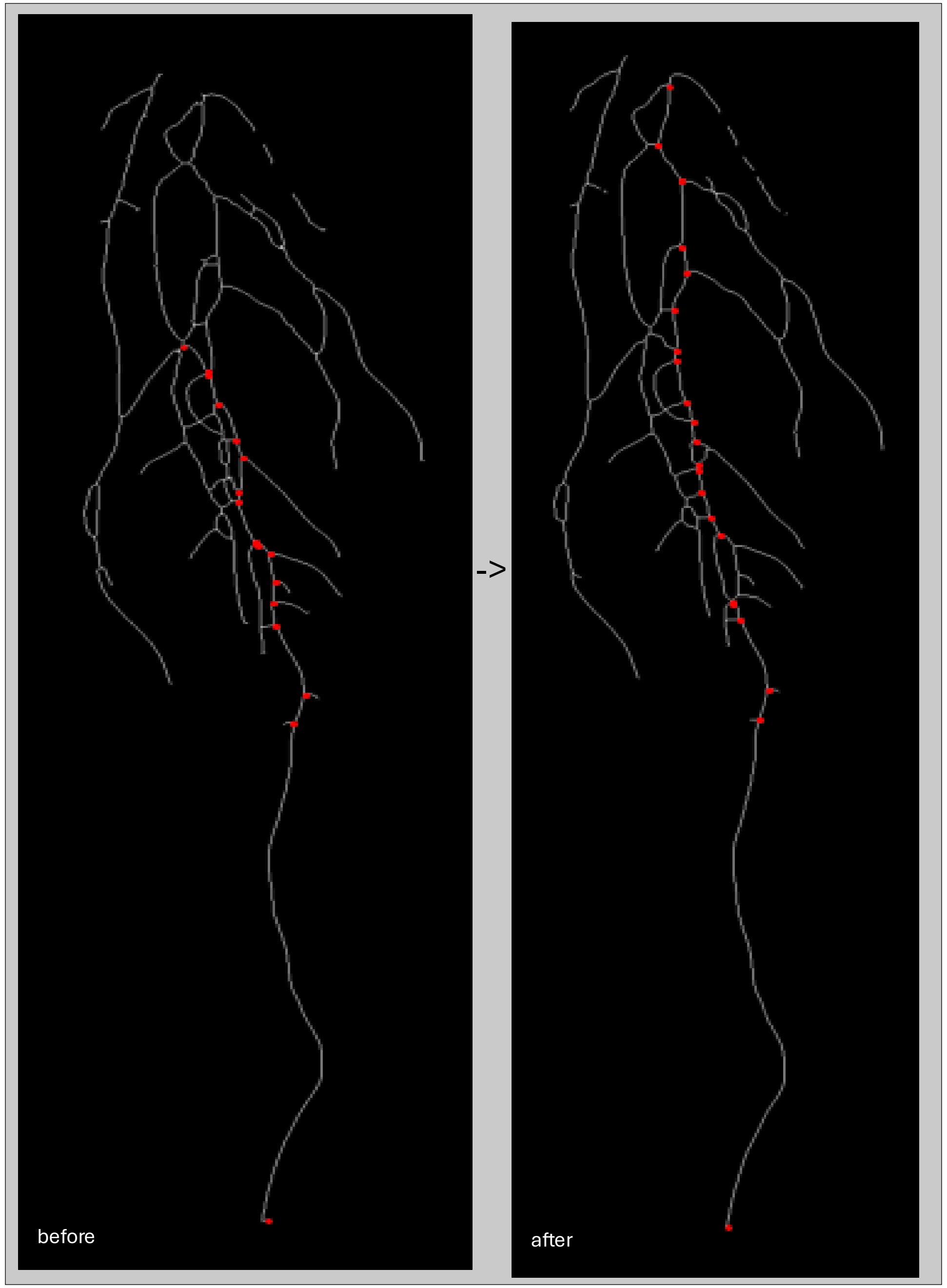
Using shoot and seed recognition to assist root segmentation







Error Analysis & Iterations



<u>Iteration 2</u>

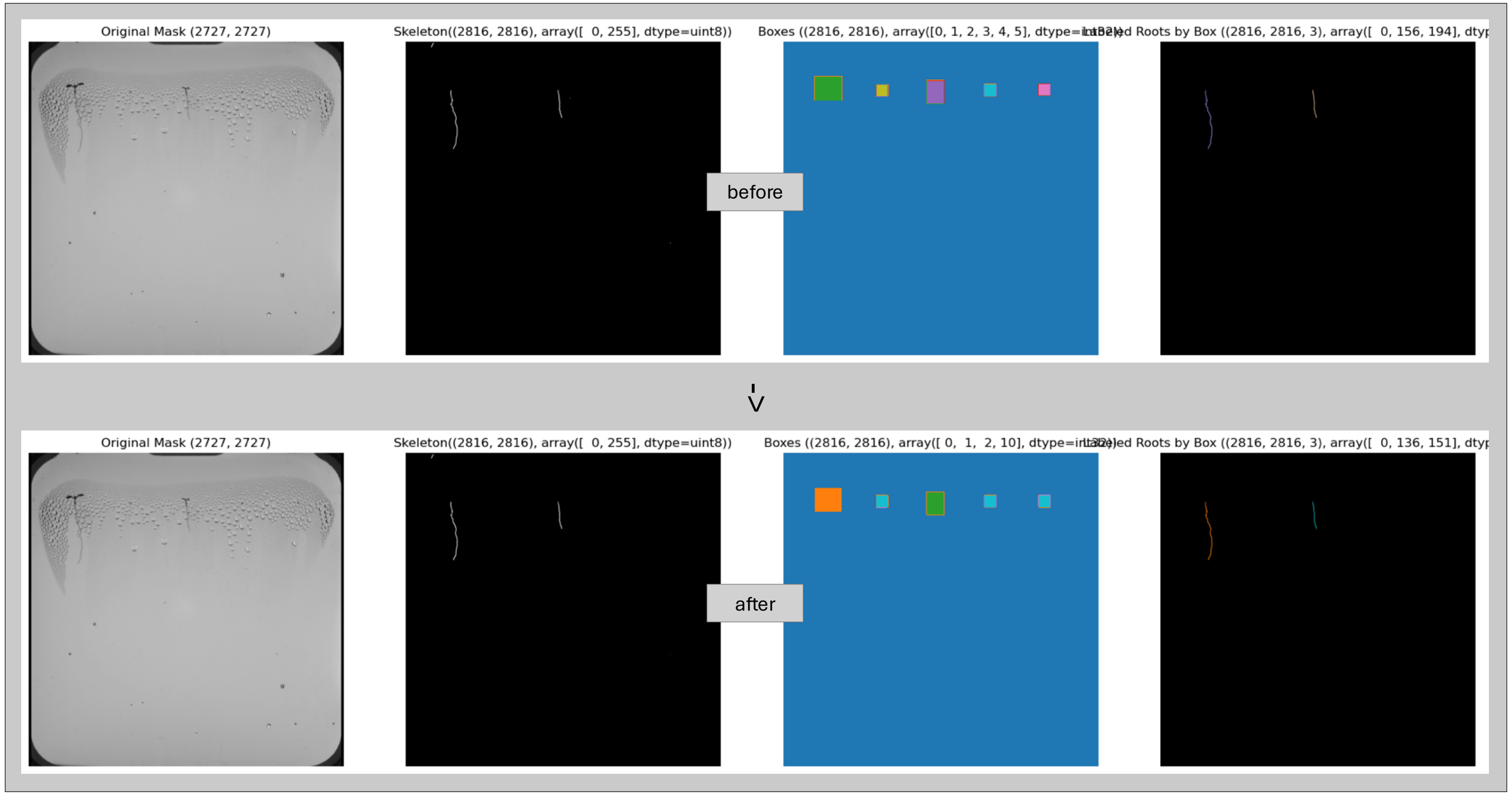
Changing primary root identification technique

The original technique consisted of searching for the biggest route of all smallest routes. Then, once we had that route, we start from the first node (the most upper node) and go down to the last one. I then implemented a function that removes all previous nodes if a node is followed by another node that is higher than itself, which wasn't working as expected.

To fix this, instead of just taking the biggest route of all smallest routes, I sorted all the routes by length in descending order, so that when a node goes up it just chooses the next biggest route within the sorted list.

This allows the algorithm to find the primary root more precisely.

Iteration 3
Changing segmentation technique



In the first approach I located individual roots/non-roots based on the segmented root mask, cutting the mask in 5 vertically and assigning from left to right. This has its limits if the roots don't go straight down, resulting to conflicting errors and not counting the last plant.

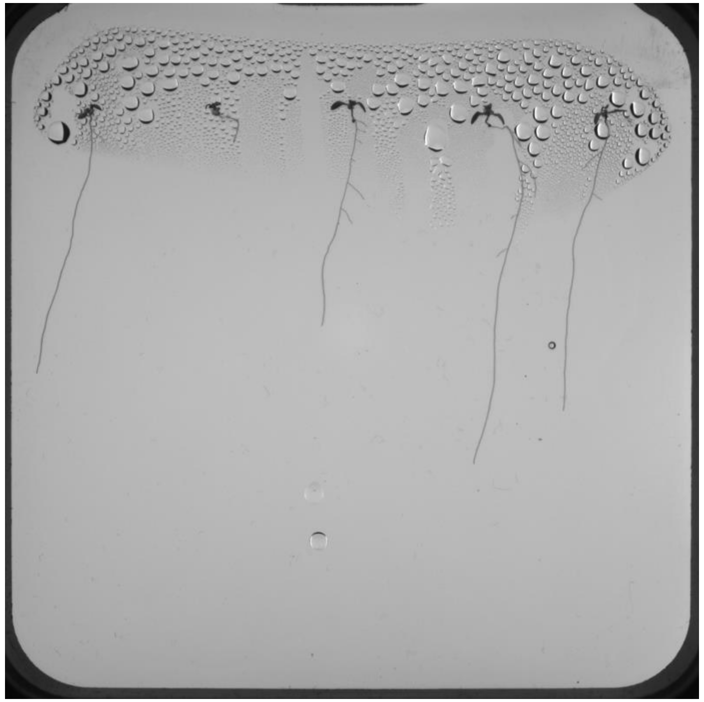
I thus changed the technique by instead locating individual roots/non-roots based on the shoot/seed boxes instead. By assigning the value 10 to boxes that touch no roots, we can use a vertical scan from left to right to classify the 5 plants/seeds.

[1, 10, 2, 10, 10]

Assumptions & Limitations

Assumptions

- Always 5 plants/seeds
- Equally aligned +/-



Limitations

- Overlapping roots (annotation?)
- Segmentation limitation (annotation?)
- Primary root identification (annotation?)
- Not enough data (small seedlings)
- Shoot/seed overlap (annotation?)

Next steps

- Annotation
- Tracking
- Flexible pipeline for tuning

Thank you for your attention