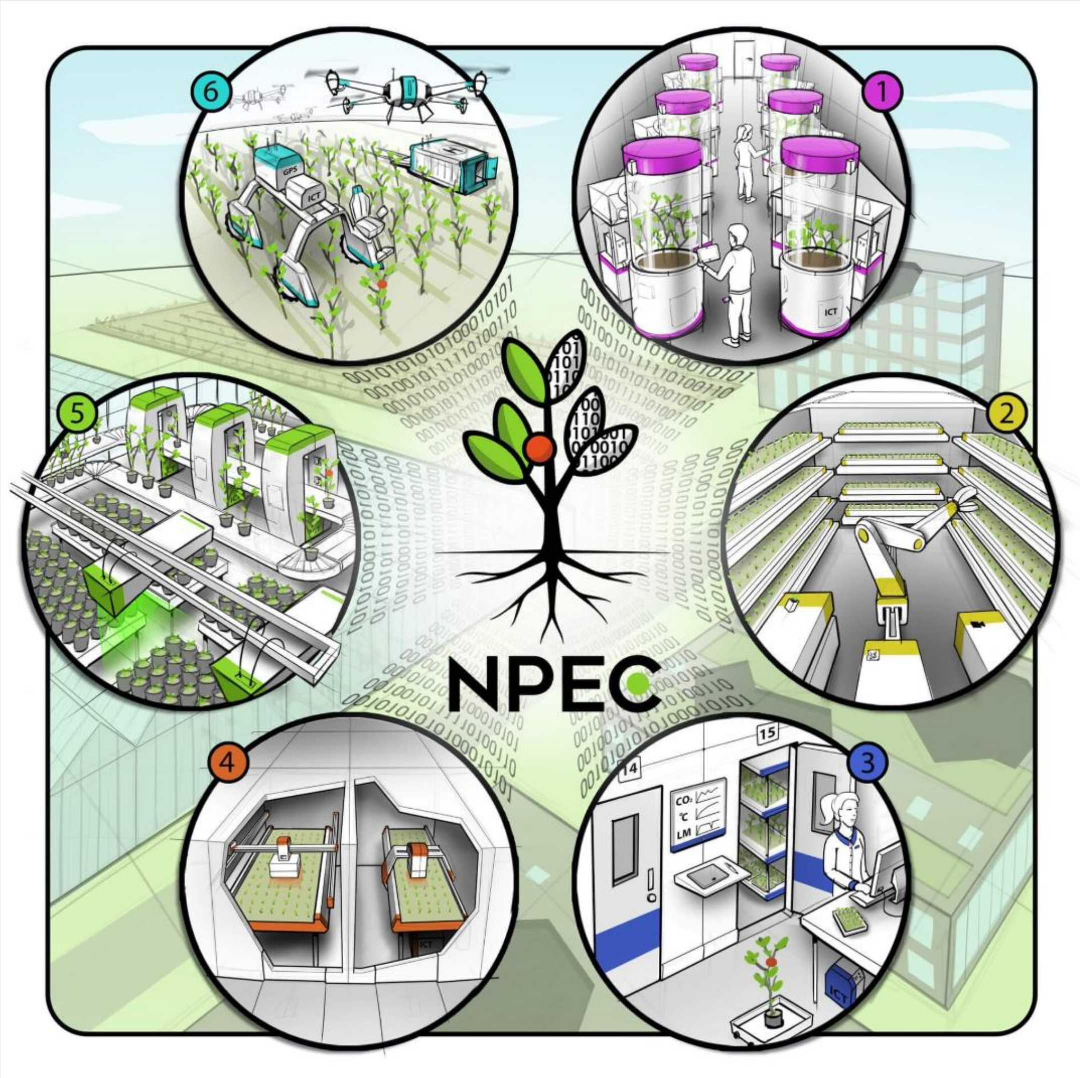
AI and the Arabidopsis thaliana

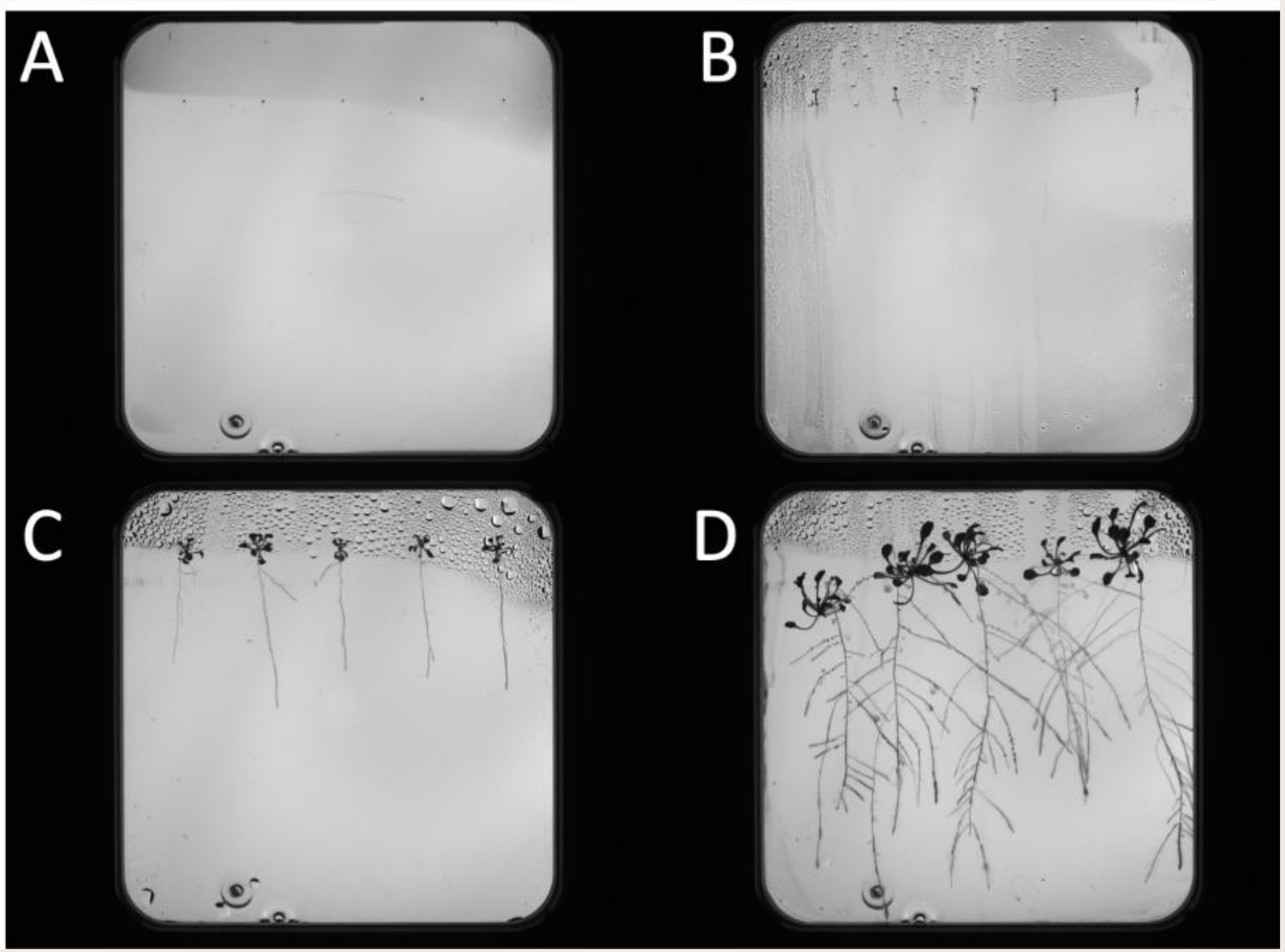
Technological solutions meet nature-based problems

Daria Elena Vlăduțu

Agenda

- Problem definition
- Solution Overview
- Results and Evaluation
- Error Analysis and Iteration
- Assumptions
- Limitations
- Next steps…





Arabidopsis thaliana
(From A to D): Seeds germinate, revealing roots, hypocotyl, and leaves. Lateral roots sprout from the primary, and they begin to crisscross

Problem definition

The challenge

Analyses of plant root images and physical interaction are manual, time-consuming, and potential sources of human error.

Core tasks

- Image Analysis
- Robotic Action

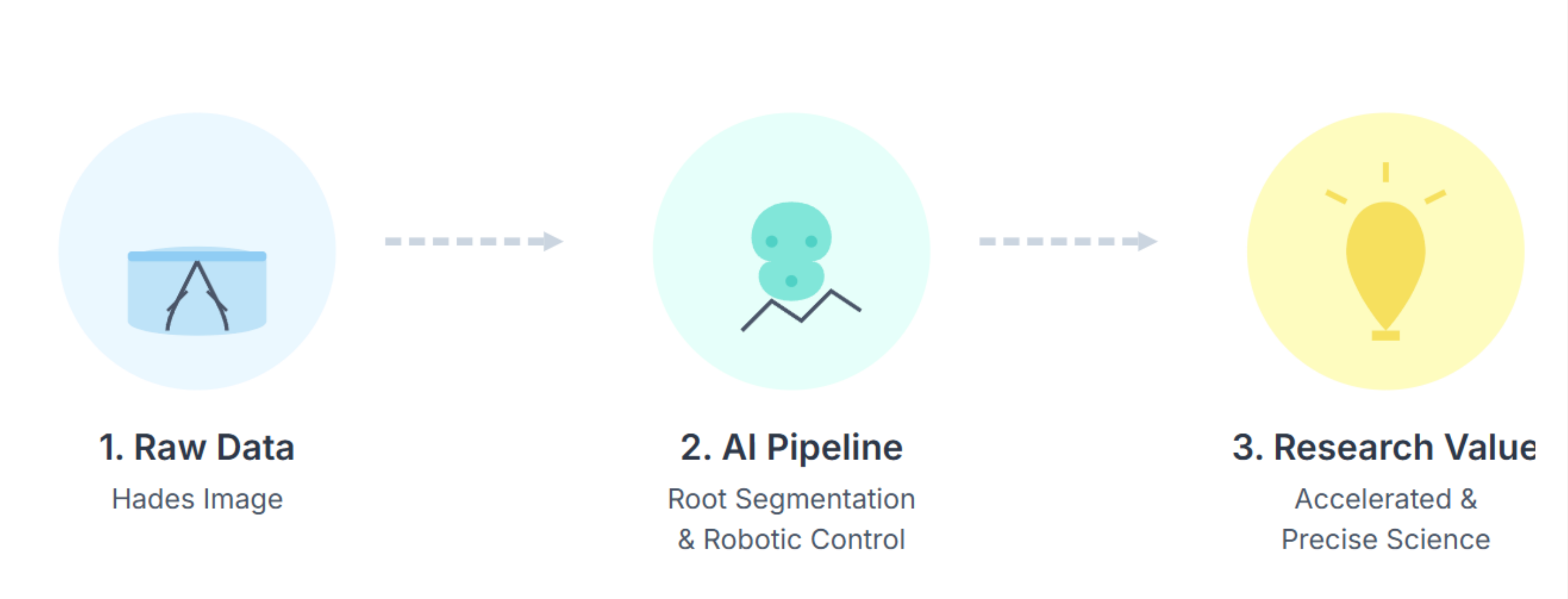
Client objectives

Automated analysis of plant root systems and robotic inoculation of plants

Solution overview

Integrating computer vision and robotics into an end-to-end inoculation pipeline

How my solution creates client value



- Directly accelerating plant science research
- High-Throughput Analysis
- Enhanced Experimental Precision

Results & Evaluation

Final model

Stats for nerds

- Custom U-net model
- Patch size: 256 x 256
- Step size: 128
- Batch size of 32
- Loss function: binary_crossentropy
- Early stopping triggered at epoch 18/30

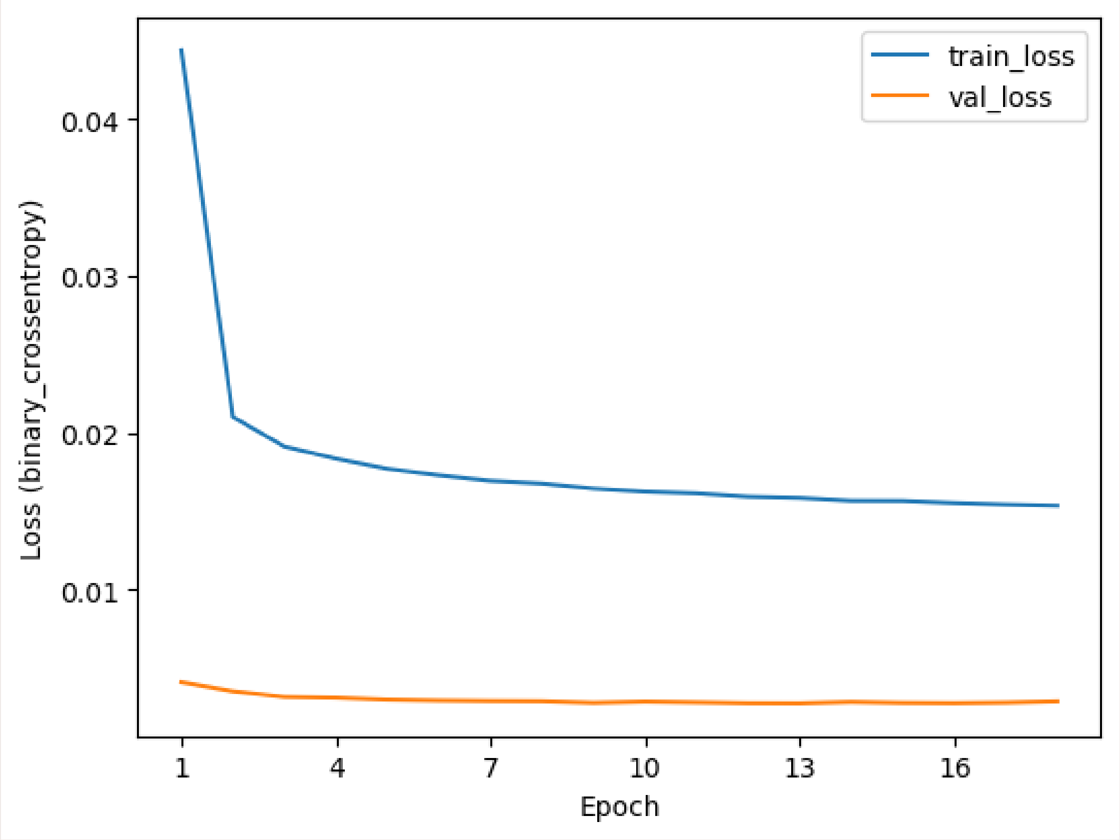
- Val_f1: 0.8374
- sMAPE: 6.073

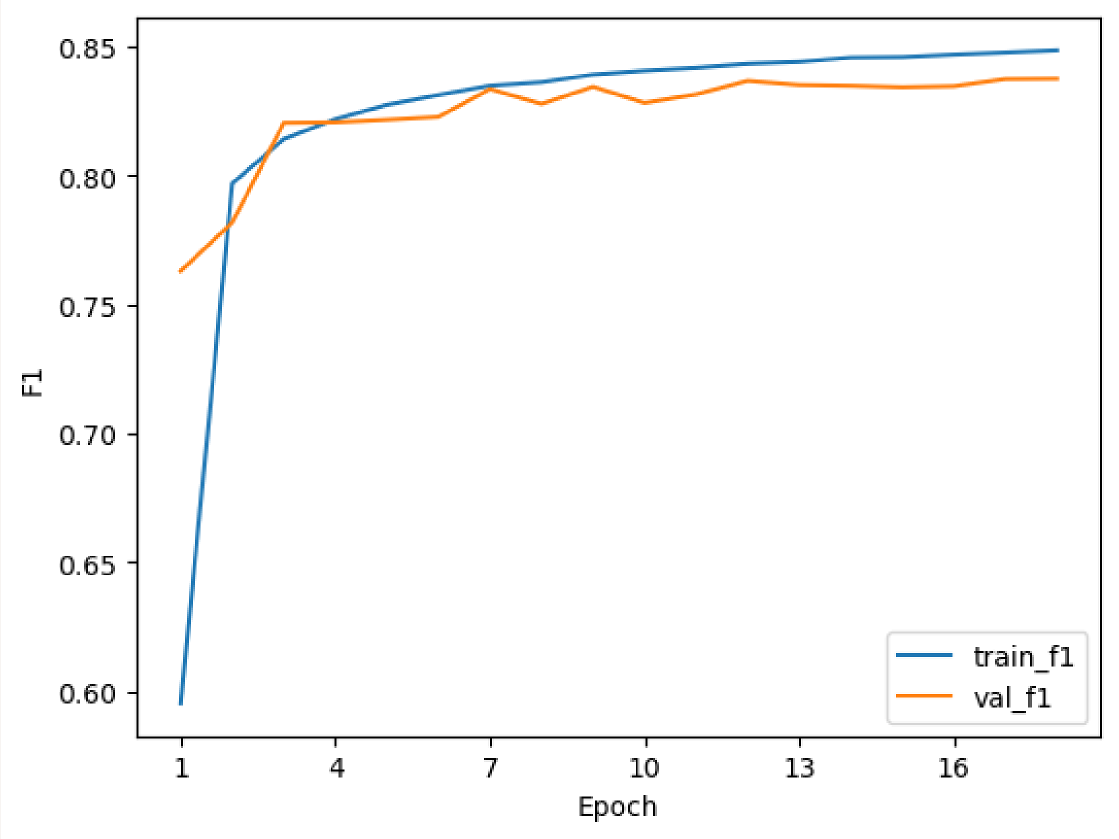
```
inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
s = inputs

# Contraction path
c1 = Conv2D(16, (3, 3), activation='relu', padding='same')(s)
c1 = Dropout(0.1)(c1)
c1 = Conv2D(16, (3, 3), activation='relu', padding='same')(c1)
p1 = MaxPooling2D((2, 2))(c1)

c2 = Conv2D(32, (3, 3), activation='relu', padding='same')(p1)
c2 = Dropout(0.1)(c2)
c2 = Conv2D(32, (3, 3), activation='relu', padding='same')(c2)
p2 = MaxPooling2D((2, 2))(c2)

c3 = Conv2D(64, (3, 3), activation='relu', padding='same')(p2)
c3 = Dropout(0.2)(c3)
c3 = Conv2D(64, (3, 3), activation='relu', padding='same')(c3)
p3 = MaxPooling2D((2, 2))(c3)

c4 = Conv2D(128, (3, 3), activation='relu', padding='same')(p3)
c4 = Dropout(0.2)(c4)
c4 = Conv2D(128, (3, 3), activation='relu', padding='same')(c4)
p4 = MaxPooling2D((2, 2))(c4)

c5 = Conv2D(256, (3, 3), activation='relu', padding='same')(p4)
c5 = Dropout(0.3)(c5)
c5 = Conv2D(256, (3, 3), activation='relu', padding='same')(c5)

# Expansive path
u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(128, (3, 3), activation='relu', padding='same')(u6)
c6 = Dropout(0.2)(c6)
c6 = Conv2D(128, (3, 3), activation='relu', padding='same')(c6)

u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), activation='relu', padding='same')(u7)
c7 = Dropout(0.2)(c7)
c7 = Conv2D(64, (3, 3), activation='relu', padding='same')(c7)

u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')(c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(32, (3, 3), activation='relu', padding='same')(u8)
c8 = Dropout(0.1)(c8)
c8 = Conv2D(32, (3, 3), activation='relu', padding='same')(c8)

u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same')(c8)
u9 = concatenate([u9, c1])
c9 = Conv2D(16, (3, 3), activation='relu', padding='same')(u9)
c9 = Dropout(0.1)(c9)
c9 = Conv2D(16, (3, 3), activation='relu', padding='same')(c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)
model = Model(inputs=[inputs], outputs=[outputs])
model.compile(optimizer='adam',
          loss="binary_crossentropy",
          metrics=['accuracy', f1])
```

U-Net: Convolutional Networks for Biomedical Image Segmentation (Ronneberger et. Al, 2015)

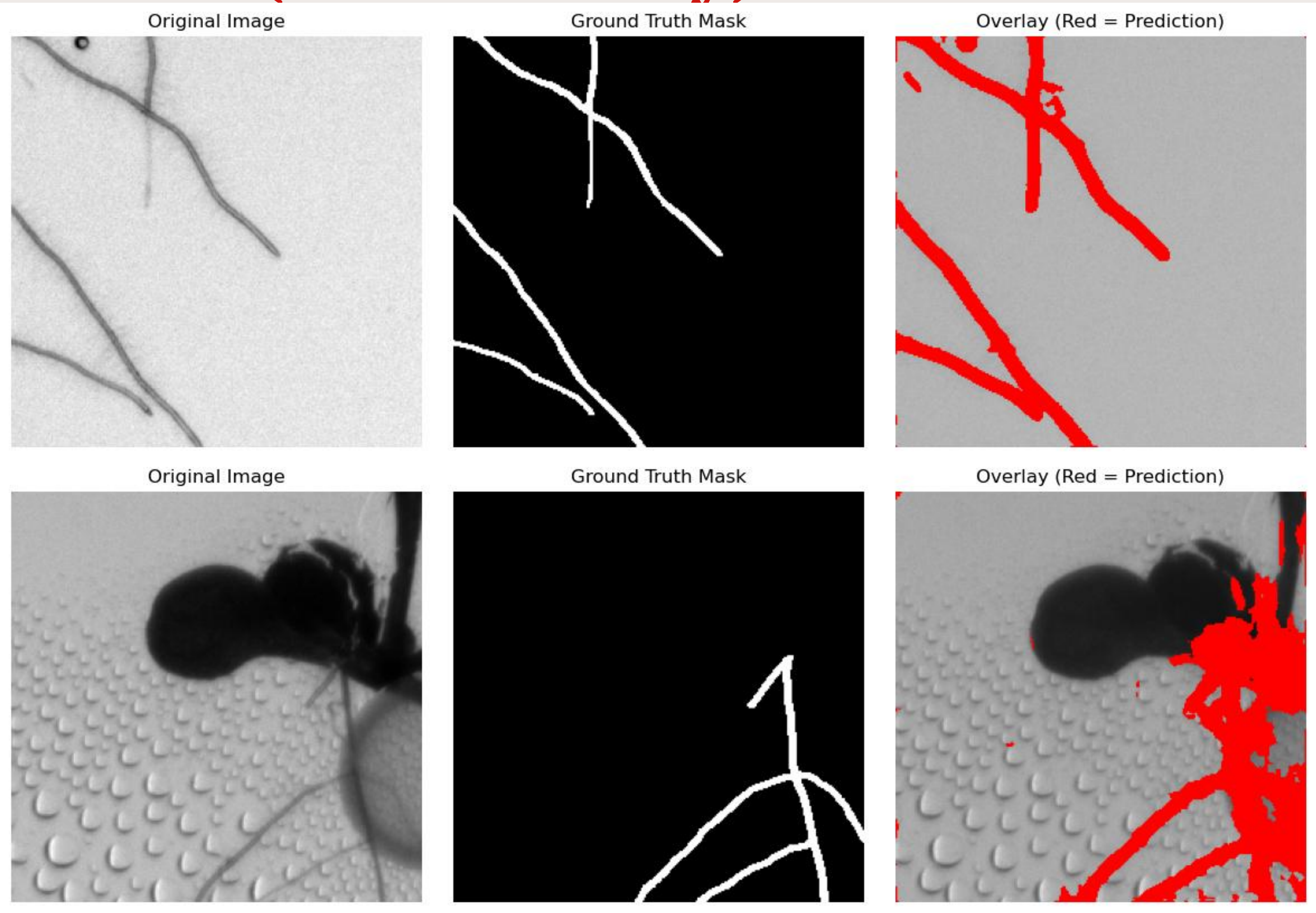
Final model

Stats for nerds

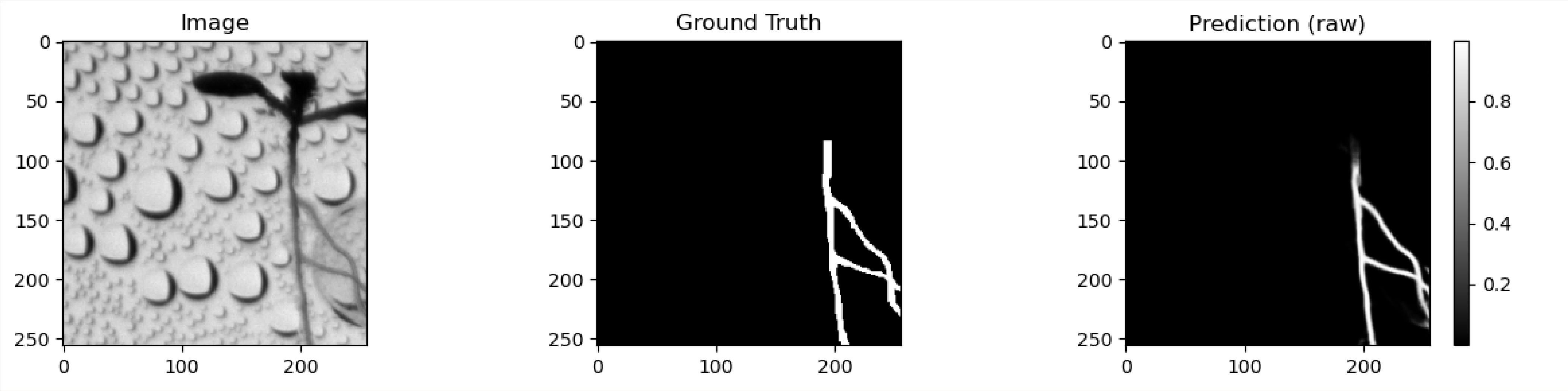


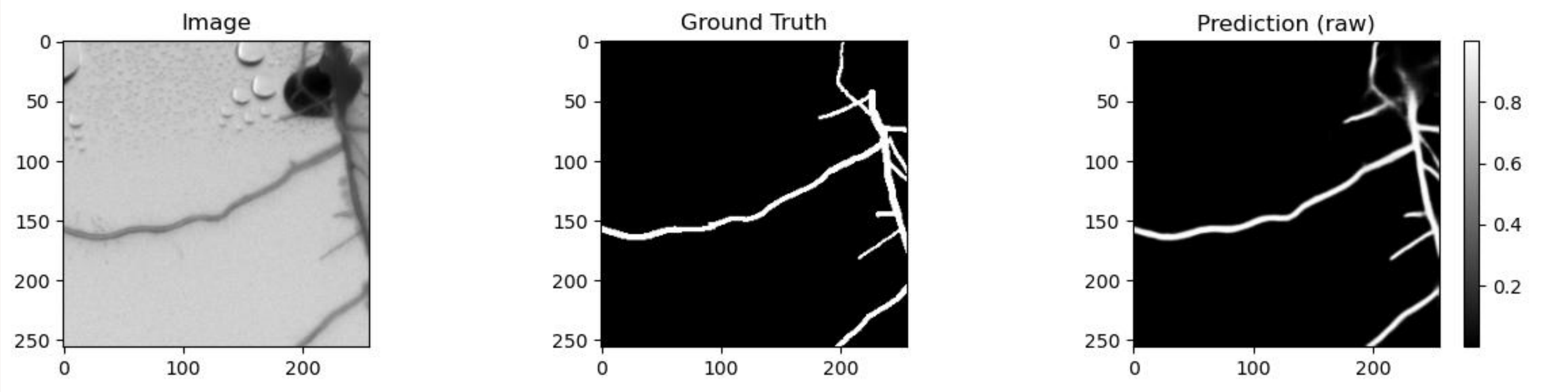


Predictions (with overlay)



Predictions (raw)





Strategy for main pipeline

Data Pre-processing & Inference	Images are padded, cropped, and patchified. Model Prediction Unpatch, unpad, uncrop to full mask
Post-processing	Thresholding at 0.10 to binarize masks Area filtering (min_area =150px) Top & Petri dish crop: removed 15% of top image and cropped to Petri dish size
Main root components	Aspect ratio threshold; min 1.5 Vertical start: max 30% Threshold of 200px for large components
Main root components	Skeltonize Function evaluates all possible paths between root tips Scoring for each path

test_image_3.tif — Double cropping action

Final Check: Original Image

Final Check: Predicted Mask Overlay

Error analysis & Iterations

Iteration 1: Experimenting with datasets

- Clean dataset
  - Merged Y2B_23 & Y2B_24
  - Removed non-root masks + unpaired images
- Simple filtered dataset
  - Kept root-rich patches (min 0.5% of patch must be root)
  - Patch includes at least one connected root segment > 75px
- Automated tuning dataset
  - Tuner finds the best min_ratio and min_area, using optuna.
  - Based on initial dataset
- Augmented dataset
  - Applied to the tuned dataset
  - Horizontal + vertical flip
  - Random rotate 90
  - Shift scale rotate
  - Random brightness contrast
- Balanced dataset
  - Used logic of tuning from automatic tuning dataset
  - Added more background patches

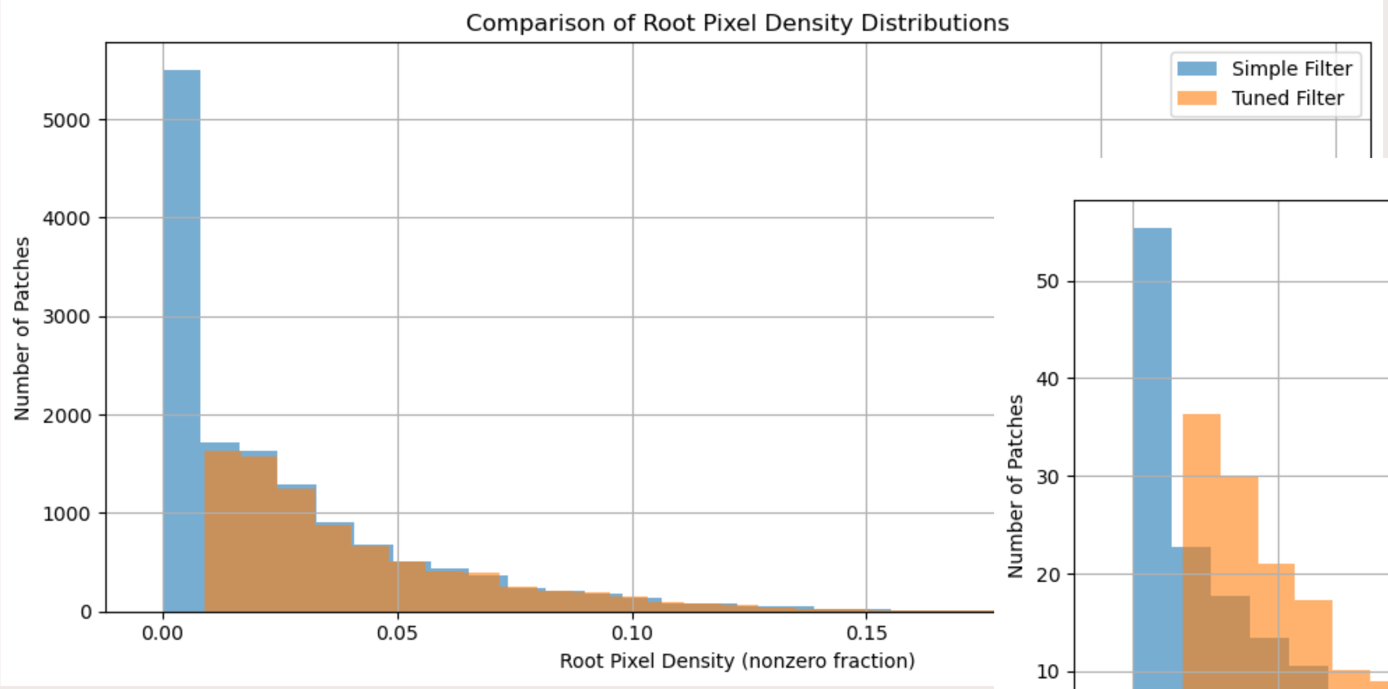
Iteration 1: Experimenting with datasets

```
Best parameters: {'min_ratio': 0.008977007432371202, 'min_area': 108}
Best score: 0.0
```

	number	value	datetime_start	datetime_complete	duration	params_min_area	params_min_ratio
0	0	0.0	2025-08-14 01:09:10.101065	2025-08-14 01:09:10.104620	0 days 00:00:00.003555	108	0.008977
1	1	0.0	2025-08-14 01:09:10.104620	2025-08-14 01:09:10.105482	0 days 00:00:00.000862	93	0.017249
2	2	0.0	2025-08-14 01:09:10.105482	2025-08-14 01:09:10.105482	0 days 00:00:00	88	0.005064
3	3	0.0	2025-08-14 01:09:10.105482	2025-08-14 01:09:10.105482	0 days 00:00:00	72	0.001108
4	4	0.0	2025-08-14 01:09:10.105482	2025-08-14 01:09:10.105482	0 days 00:00:00	82	0.010496
5	5	0.0	2025-08-14 01:09:10.105482	2025-08-14 01:09:10.113200	0 days 00:00:00.007718	45	0.001232
6	6	0.0	2025-08-14 01:09:10.114232	2025-08-14 01:09:10.115239	0 days 00:00:00.001007	133	0.004415
7	7	0.0	2025-08-14 01:09:10.115239	2025-08-14 01:09:10.116243	0 days 00:00:00.001004	81	0.008460
8	8	0.0	2025-08-14 01:09:10.117246	2025-08-14 01:09:10.117246	0 days 00:00:00	25	0.002952
9	9	0.0	2025-08-14 01:09:10.118251	2025-08-14 01:09:10.118251	0 days 00:00:00	128	0.007016
10	10	0.0	2025-08-14 01:09:10.119253	2025-08-14 01:09:10.120633	0 days 00:00:00.001380	150	0.018274
11	11	0.0	2025-08-14 01:09:10.120633	2025-08-14 01:09:10.131915	0 days 00:00:00.011282	112	0.019896
12	12	0.0	2025-08-14 01:09:10.132920	2025-08-14 01:09:10.135651	0 days 00:00:00.002731	107	0.012281
13	13	0.0	2025-08-14 01:09:10.135651	2025-08-14 01:09:10.140962	0 days 00:00:00.005311	103	0.013945
14	14	0.0	2025-08-14 01:09:10.140962	2025-08-14 01:09:10.149595	0 days 00:00:00.008633	61	0.006711
15	15	0.0	2025-08-14 01:09:10.149595	2025-08-14 01:09:10.153547	0 days 00:00:00.003952	98	0.003105
16	16	0.0	2025-08-14 01:09:10.153547	2025-08-14 01:09:10.153547	0 days 00:00:00	123	0.013749
17	17	0.0	2025-08-14 01:09:10.153547	2025-08-14 01:09:10.166402	0 days 00:00:00.012855	58	0.009526
18	18	0.0	2025-08-14 01:09:10.167407	2025-08-14 01:09:10.170842	0 days 00:00:00.003435	146	0.001818
19	19	0.0	2025-08-14 01:09:10.170842	2025-08-14 01:09:10.177595	0 days 00:00:00.006753	117	0.016295
20	20	0.0	2025-08-14 01:09:10.177595	2025-08-14 01:09:10.185496	0 days 00:00:00.007901	95	0.005291
21	21	0.0	2025-08-14 01:09:10.185496	2025-08-14 01:09:10.190328	0 days 00:00:00.004832	88	0.004101
22	22	0.0	2025-08-14 01:09:10.191328	2025-08-14 01:09:10.197742	0 days 00:00:00.006414	72	0.006415
23	23	0.0	2025-08-14 01:09:10.197742	2025-08-14 01:09:10.202043	0 days 00:00:00.004301	92	0.003108
24	24	0.0	2025-08-14 01:09:10.202043	2025-08-14 01:09:10.202043	0 days 00:00:00	72	0.008529
25	25	0.0	2025-08-14 01:09:10.202043	2025-08-14 01:09:10.217728	0 days 00:00:00.015685	137	0.010964
26	26	0.0	2025-08-14 01:09:10.217728	2025-08-14 01:09:10.220631	0 days 00:00:00.002903	115	0.005015
27	27	0.0	2025-08-14 01:09:10.225341	2025-08-14 01:09:10.230319	0 days 00:00:00.004978	106	0.002173
28	28	0.0	2025-08-14 01:09:10.231330	2025-08-14 01:09:10.237126	0 days 00:00:00.005796	50	0.014634
29	29	0.0	2025-08-14 01:09:10.238129	2025-08-14 01:09:10.243970	0 days 00:00:00.005841	81	0.007712

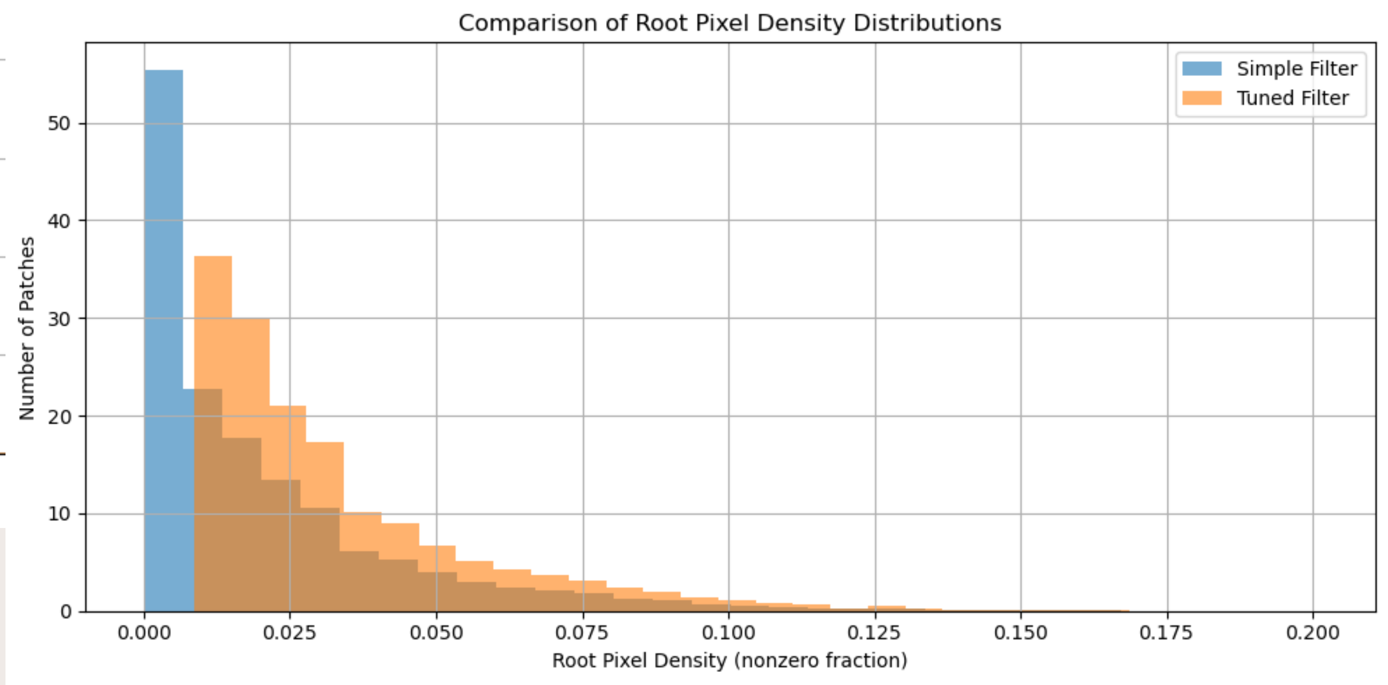
Iteration 1: Experimenting with datasets

2023 dataset



Mean density (simple): 0.0251
Mean density (tuned) : 0.0409

2024 dataset



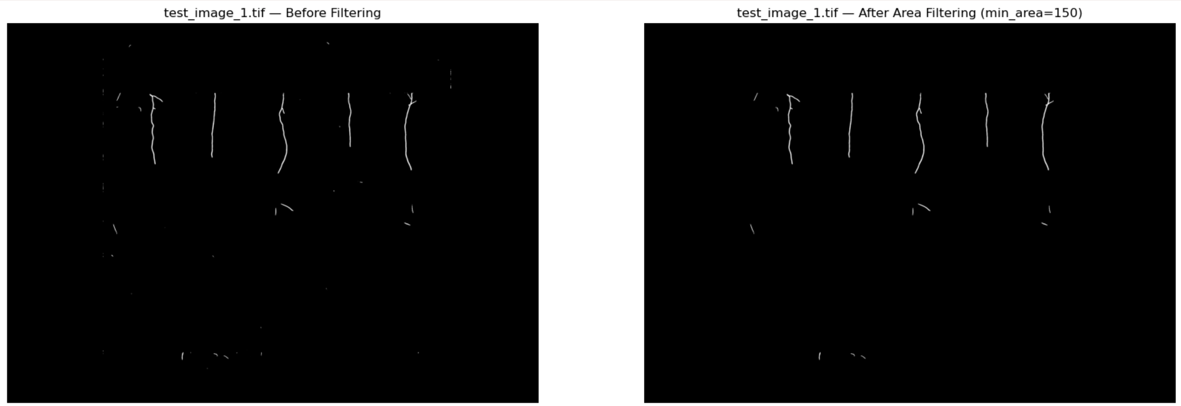
Mean density (simple): 0.0198
Mean density (tuned) : 0.0329

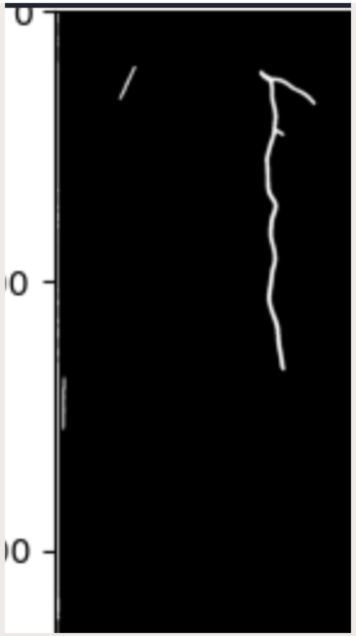
Iteration 2: Post-processing of masks

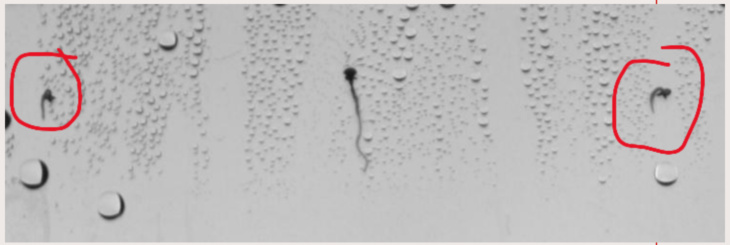
- Issue: Salt-and-pepper noise
- Initial solution: Morphological closing
- Final solution: Filtering by area

- Issue: First plant was being mistaken for a thin white line along the edge
- Initial solution: Use a circular crop
- Final solution: Have a 'tighter' crop

- Issue: Newly germinated seeds were being missed entirely.
- Solution: Lower filtering threshold from 250px to 50px

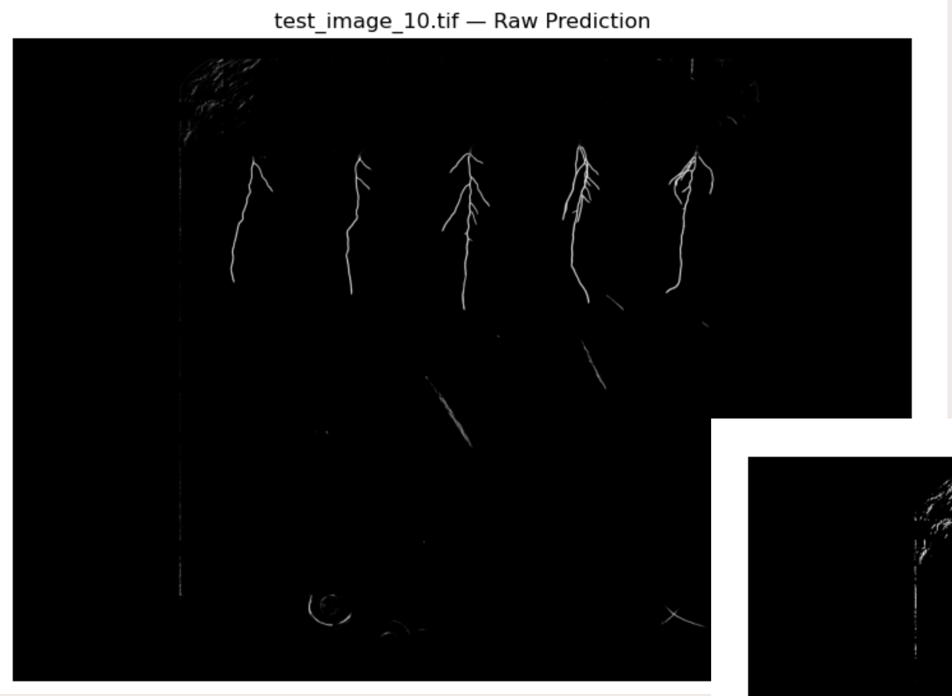


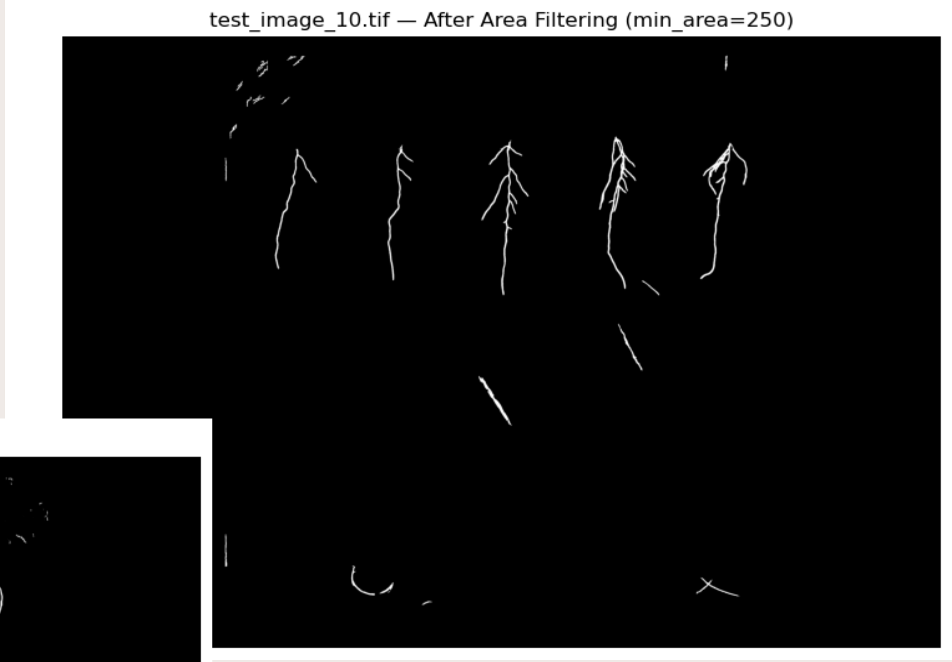


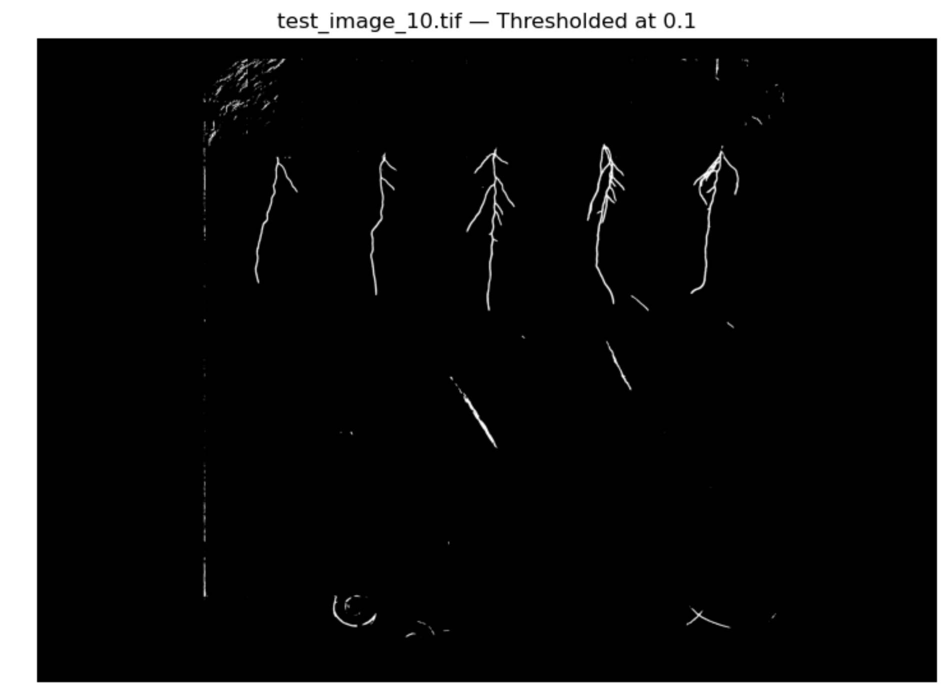


Iteration 2: Post-processing of masks

Examples from another trial

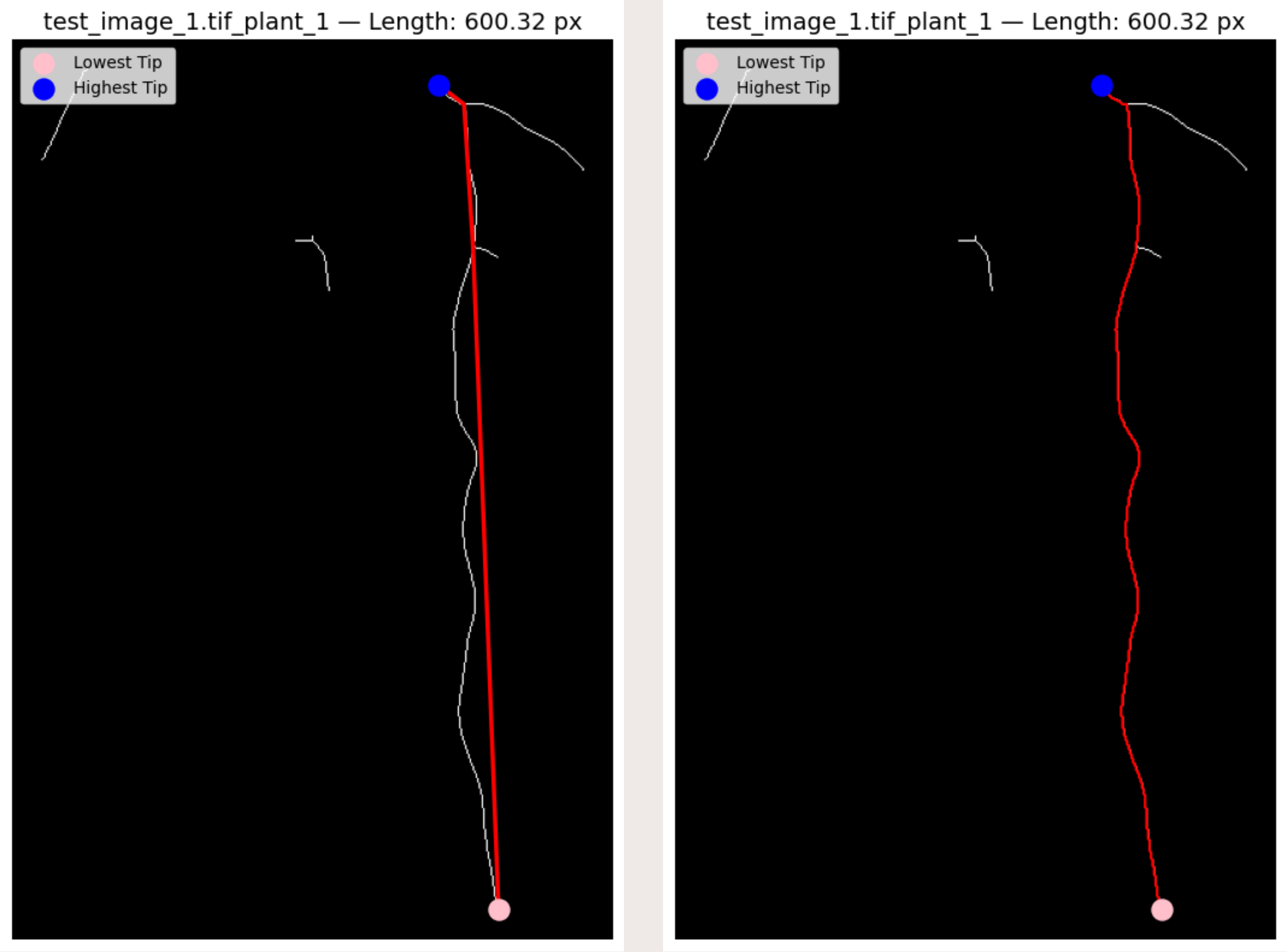






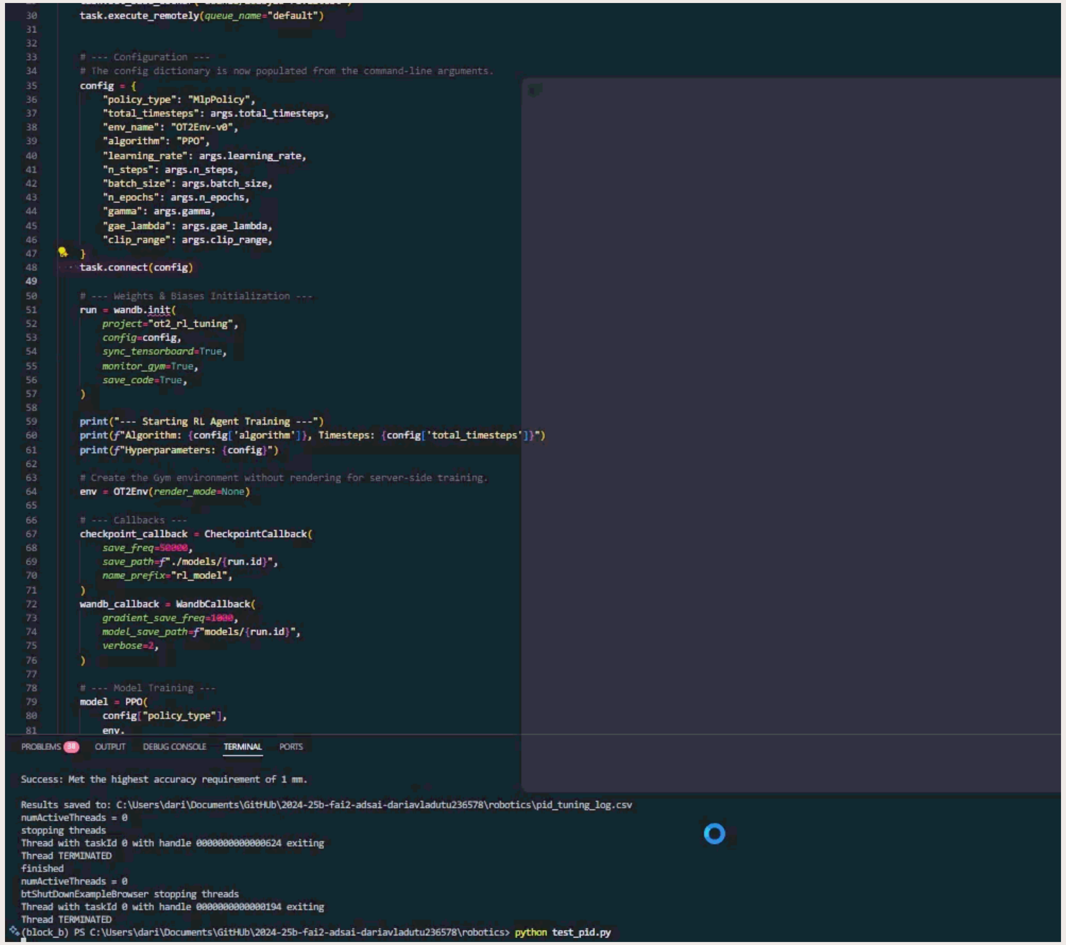
Iteration 3: Root length estimation

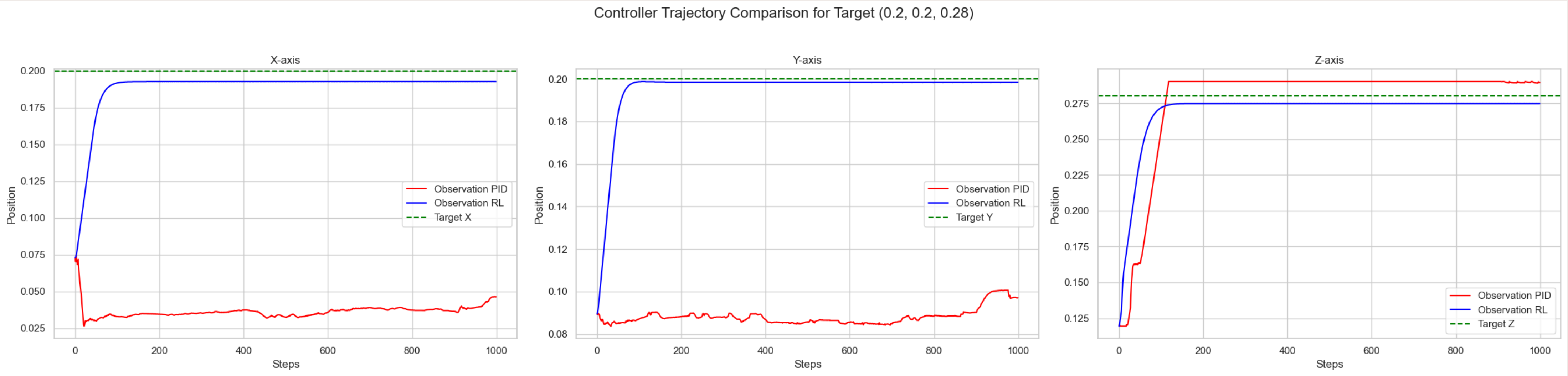
- Issue: Find the primary root length
- Initial solution: Scoring system between top/bottom tip nodes length, verticality and centrality.
- Proposed solution: The A-star search



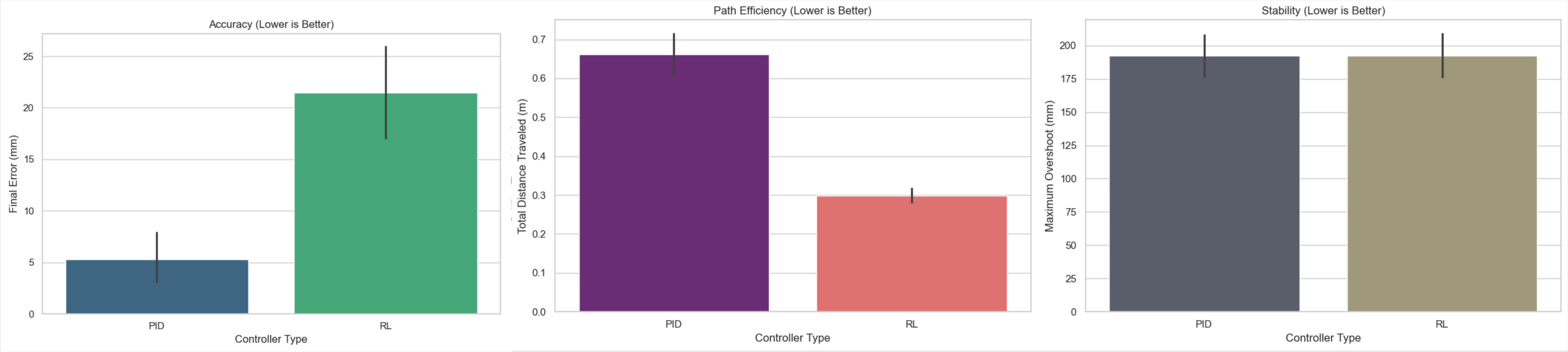
Robotics

- PID controller:
  - Best gains:
    - KP = 5.0
    - KI = 0.5
    - KD = 2.0
  - Final error: 0.592mm
- Reinforcement learning journey:
  - 4 official iterations
  - Continuous experimentation
  - Final error of RL agent: 9.2mm





Controller Trajectory Comparison for Target (0.2, 0.2, 0.28)
X-axis
Y-axis
Z-axis
Position
Steps
Observation PID
Observation RL
Target X
Target Y
Target Z



Accuracy (Lower is Better)
Final Error (mm)
Path Efficiency (Lower is Better)
Total Distance Traveled (m)
Stability (Lower is Better)
Maximum Overshoot (mm)
PID
RL
Controller Type

Assumptions

Assumptions made during the design of the pipeline

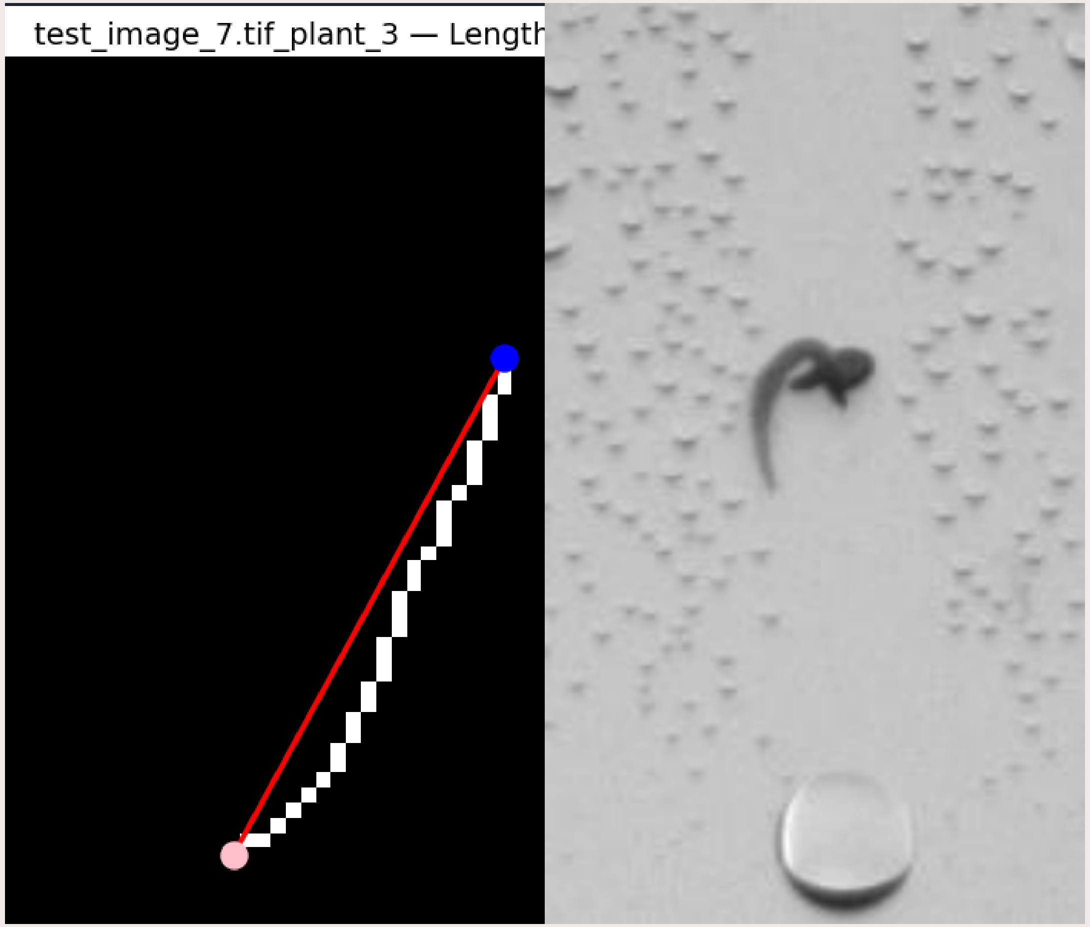
Top 15% of image height is not useful data

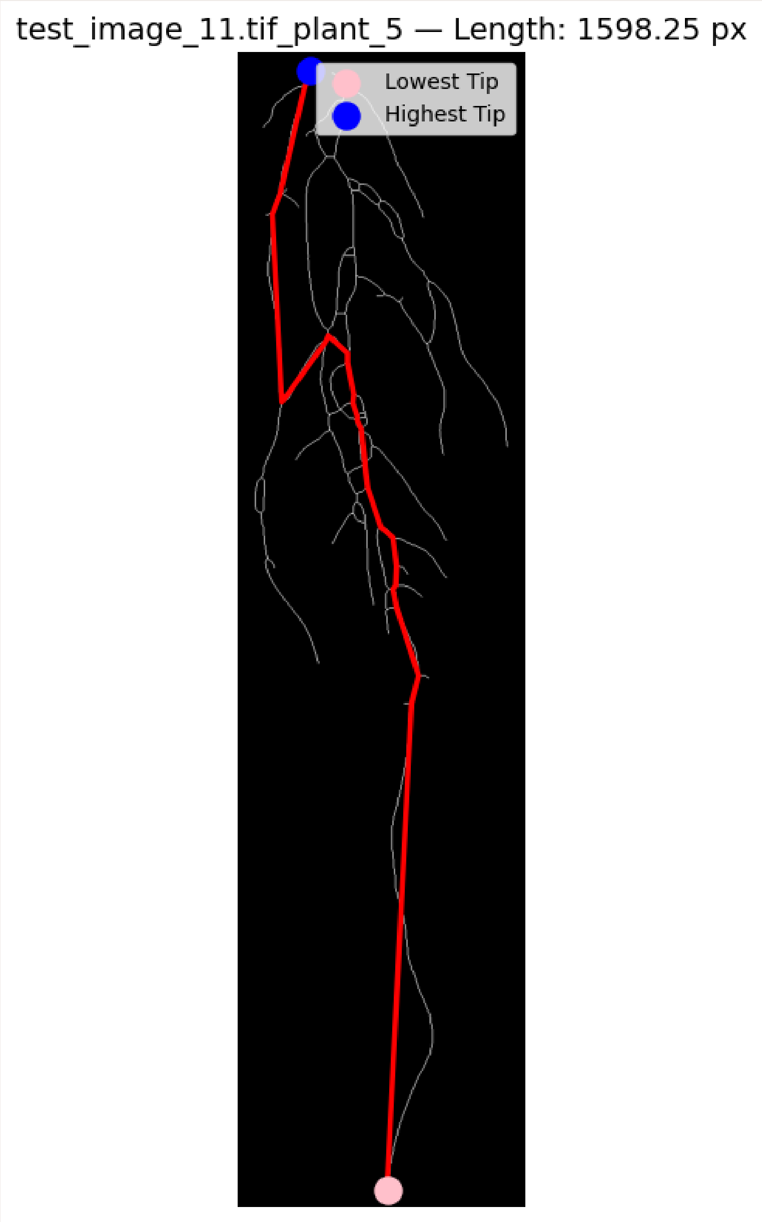
Components with an area < 150px are noise

Roots are objects that are tall and skinny (area ratio threshold > 1.5)

The primary root is the shortest path from the **topmost to bottommost** node

Limitations





Dataset limitations

Model cannot properly detect smaller roots

Overlapping and dense roots make it hard for the model to identify the main root path accurately

Next steps

What can be done?

1

A faster, more computationally proficient computer to train a model, using a mega dataset.

2

A dataset containing more newly germinated seeds.

3

A dataset containing manually annotated main roots

Thank you