NPEC Report: Plant root inoculation with Computer Vision Robotics and AI

Francisco Mansilha 220387

DISCOVER YOUR WORLD

Academy/Department

Saturday 25 January 2024

Breda University OF APPLIED SCIENCES

NPEC Report: Plant root inoculation with Computer Vision Robotics and AI

Francisco R. Mansilha

Breda University of Applied Sciences

2023-24 Y2B Project 2B ADS&AI, Applied Data Science and Artificial Intelligence

Dean van Aswegen, MSc. & Vinicius Munaldi Lube, PhD.

January 25, 2024

Contents

1.1 Introduction

This report presents a project developed for the Netherlands Plant Eco-phenotyping Centre (NPEC), aimed at innovating plant science through the integration of computer vision, reinforcement learning, and robotics. The project focuses on segmenting plant roots from images and automating inoculation processes, critical for enhancing our understanding of plant phenotypes. These phenotypes, which encompass morphological, physiological, and biochemical traits, are pivotal for advancing sustainable agriculture and meeting future food and material needs without compromising our planet.

NPEC, a leader in plant phenotyping, operates sophisticated facilities for high-resolution, high-throughput data collection on plant characteristics. The project specifically utilizes the Hades system within the Plant-Microbe Interaction Phenotyping module, designed for in-vitro root system analysis. This system facilitates non-destructive, automated assessment of plant roots, playing a crucial role in understanding plant growth and interactions with microbes under controlled conditions.

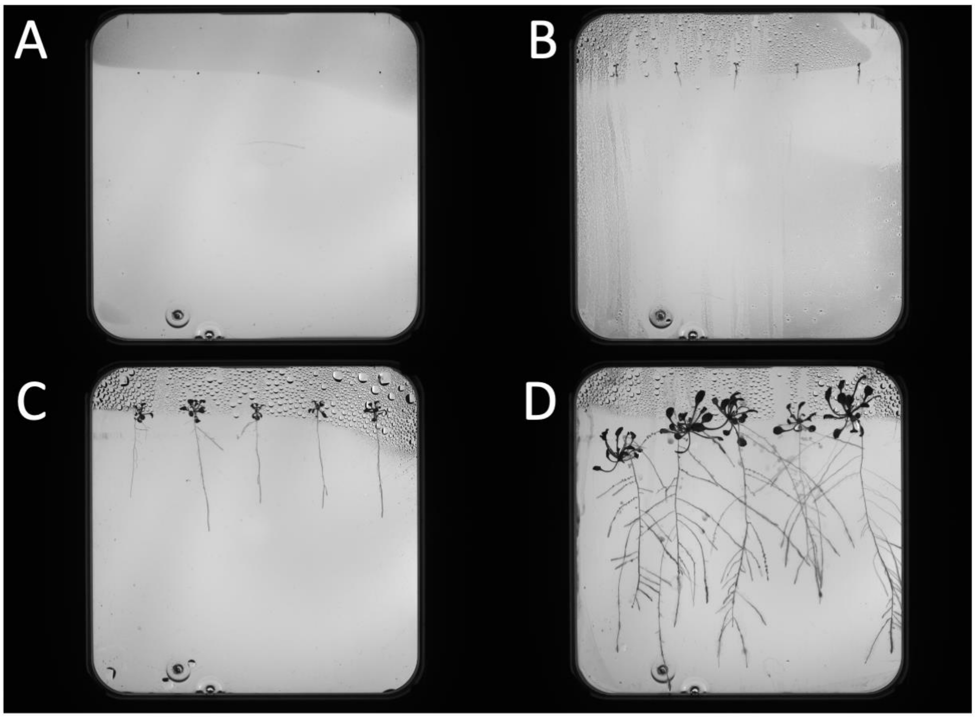
The solution employs a combination of advanced computer vision and robotics. A computer vision pipeline, incorporating traditional and AI techniques which performs tasks from region of interest extraction to semantic segmentation and morphometric analysis. Concurrently, robotics, particularly the manipulation of the Opentron OT-2 liquid handling robot, utilizes reinforcement learning and PID control for precise plant inoculation. This dual approach allowed us to present an effective solution to the challenge of automating the inoculation process of plant roots.

1.2 Dataset Description

The dataset central to this project comprises images of Arabidopsis thaliana, a model organism in plant biology, grown in Petri dishes. Initially, five seeds of Arabidopsis thaliana are sown in each dish (Figure 1A) and photographed daily. As days go by, seeds germinate, revealing roots, hypocotyl, and leaves (Figure 1B). Eventually, smaller lateral roots sprout from the primary (also called main) root (Figure 1C), and over time, these roots begin to crisscross (Figure 1D).

Figure 1

Sample images from the dataset



According to Valérian Méline, a Phenomics and Data Scientist at NPEC, the photographing process is automated by using a robotic arm developed by Photon Systems Instruments, which takes the Petri Dish and places it in a photography chamber (designed to minimize external light and reflections) and it is then photographed with a single camera.

These images track the growth progression of Arabidopsis from seed sowing to the emergence and intermingling of roots. Utilizing the Hades system, part of the Plant-Microbe Interaction Phenotyping Module, the dataset provides high-throughput phenotyping capabilities, enabling the detailed characterization of in-vitro root systems under controlled conditions. This approach ensures comprehensive data on root architecture, benefiting from automated processes such as seed sowing, vernalization, and microbial treatments, alongside hyperspectral imaging for dynamic assessments.

The dataset can be further divided into 3 sub-datasets:

1. Segmentation dataset, 126 images without any labels.
2. Measurement dataset, 3 images with landmark locations, primary and lateral root lengths.
3. Kaggle dataset, 11 images with primary root length measurements.

1.3 Potential Dataset Bias

It's important to clarify the potential bias that this dataset encompasses.

1.3.1 Selection Bias

Like biases in datasets highlighted by Torralba et al. (2011), the dataset's focus on Arabidopsis thaliana exclusively may limit the generalizability of the findings to other plant species, akin to how datasets with images of cars from specific perspectives may not perform well on cars viewed from uncommon angles.

1.3.2 Capture Bias

The automated photographing process, while minimizing human error, introduces its own form of bias. The presence of scratches on Petri dishes, water droplets, seeds fallen to the bottom of the petri dish due to the robotic arm handling, and microbial growth, misidentified by models as part of the plant root system, mirrors the capture bias seen in image datasets where the angle and context of photos influence model performance. This reflects capture bias, where the conditions under which data is collected influence model outcomes. In Uncovering Bias in the PlantVillage Dataset Noyan (2022), a similar issue was demonstrated, where background noise significantly impacted model accuracy, specifically a machine learning model using only 8 pixels from the PlantVillage image backgrounds achieved 49.0% accuracy on the held-out test set, when the random guessing accuracy is of 2.6%. This suggests that even seemingly minor dataset features can lead to substantial biases.

Figure 2

Example of scratches in the petri dish

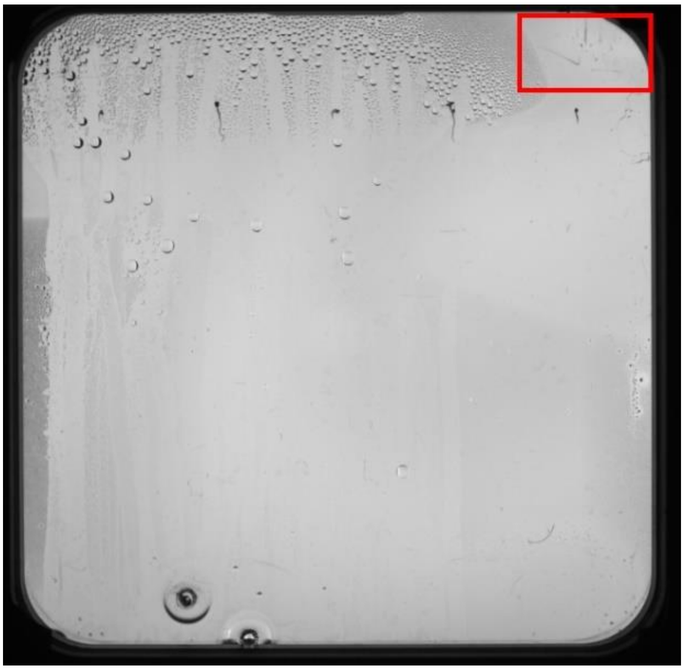
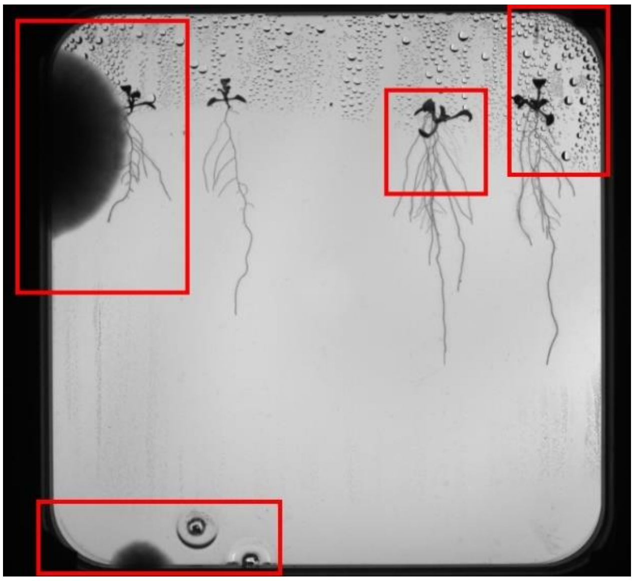


Figure 3

Examples of bacterial growth, water drops and shoot overlapping root



1.3.3 Label Bias

The project's dataset includes meticulous labelling for five plant classes: root, shoot, seed, occluded root (created specifically for cases where roots are hidden behind other plant organs like leaves or obscured by artifacts like bubbles), and background (formed by summing and inverting all the other masks).

However, the labelling process involved multiple individuals, and at the time, there was no consensus on what should precisely qualify as an occluded root. These inconsistencies in labelling have the potential to impact the performance of our models.

Dealing with challenges such as overlapping root tips and obscured plant sections in labelling is reminiscent of issues discussed in the paper Unbiased Look at Dataset Bias Torralba et al. (2011). This highlights the significance of accurate, bias-aware labelling in datasets to prevent misleading outcomes during model training. For example, when background noise correlates with labels, deep learning models can exploit this bias, affecting their predictions.

1.3.4 Environmental and Treatment Bias

The controlled conditions under which the dataset was collected might not fully represent the variability encountered in natural environments, introducing a bias towards laboratory conditions. This echoes concerns in Uncovering Bias in the PlantVillage Dataset Noyan (2022), where models trained on data collected under uniform conditions struggled with more variable real-world data.

1.4 Datalab Tasks

To complete the project, DataLab tasks were created to divide the necessary work into sections which build upon each other and end up delivering a solution. Here is a description of said tasks:

1.4.1 Task 1 Image Annotation

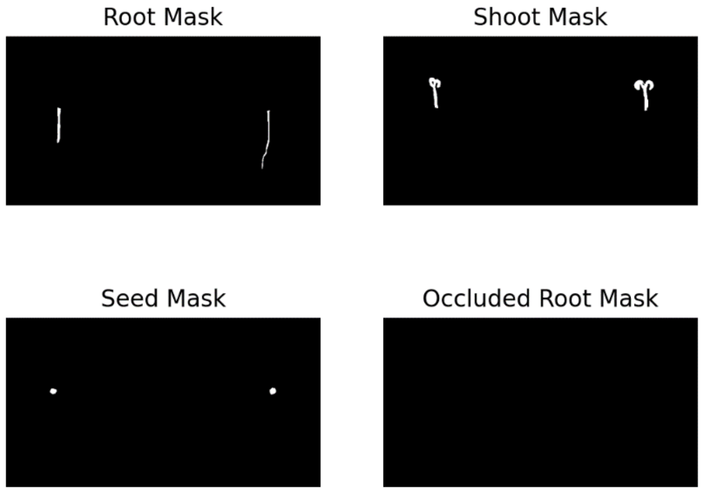
Image annotation in computer vision involves marking specific regions or features within an image to provide additional information. For this project, the goal of this task was to segment 2 or 3 plant images into four classes corresponding to the root, shoot, seed, and occluded root (for cases where roots are occluded behind other plant organs or artifacts such as leaves and bubbles).

Said masks followed a specific naming convention and met the requirements of maintaining the original image shape, being binary files and passing through a quality control check.

I labelled 2 images which both passed all the requirements and quality control check.

Figure 4

Examples of the labelled classes



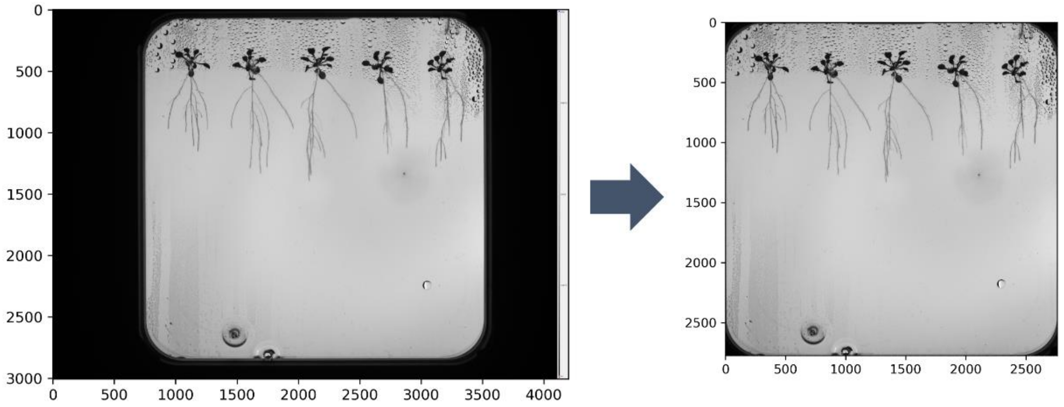
1.4.2 Task 2 Region of Interest (ROI) Extraction

ROI Extraction involves identifying and isolating specific parts of an image deemed important for further analysis. In this project, the ROI is the Petri Dish. Using traditional computer vision, we have written code that crops the Petri dish from any raw image from the dataset.

For this task the client requirement was that the Detected Petri dish edges must be within ± 15 pixels of the actual Petri dish edges. My solution was able to meet the required goal and mentioned requirements.

Figure 5

Examples of Petri dish extraction



1.4.3 Task 3 Instance Segmentation

Instance segmentation is the task of classifying and delineating each individual object (i.e. instance) in an image at the pixel level. For this task, we performed instance segmentation for five plants. In my solution, I used only traditional computer vision techniques such as applying thresholding to the images and connected components to successfully segmented each plant.

Figure 6

Examples of Instance segmentation

1.4.4 Task 4 Semantic Segmentation

For this task we were expected to train one or multiple deep-learning models for semantic segmentation. Specifically, using the labelled dataset obtained from task 1, we were expected to create models capable of segmenting five classes. These classes correspond to the ones mentioned previously (root, shoot, occluded root, and seed) as well as a new class, background (the labels for which were obtained by combining and inversing the labels of the other 4 classes).

The client requirements were that the root model incorporated at least 0.5 Intersection over Union (IoU) on the test set (a subset of the segmentation dataset.

For my solution to this task, I created 6 models. One to segment each class individually and another one for multiclass segmentation. For concision purposes I will focus only on the root model and its performance metrics as it proved to be the most useful for the remainder of the project. The model used the U-net architecture, horizontal flip data augmentation and a batch size of 32. It attained the following metrics on the test set:

- Test Loss: 0.006946456618607044
- Test Accuracy: 0.9973463416099548
- Test F1 Score: 0.8191097974777222
- Test IoU: 0.9028586745262146

Figure 7

Plot of the training and validation IoU

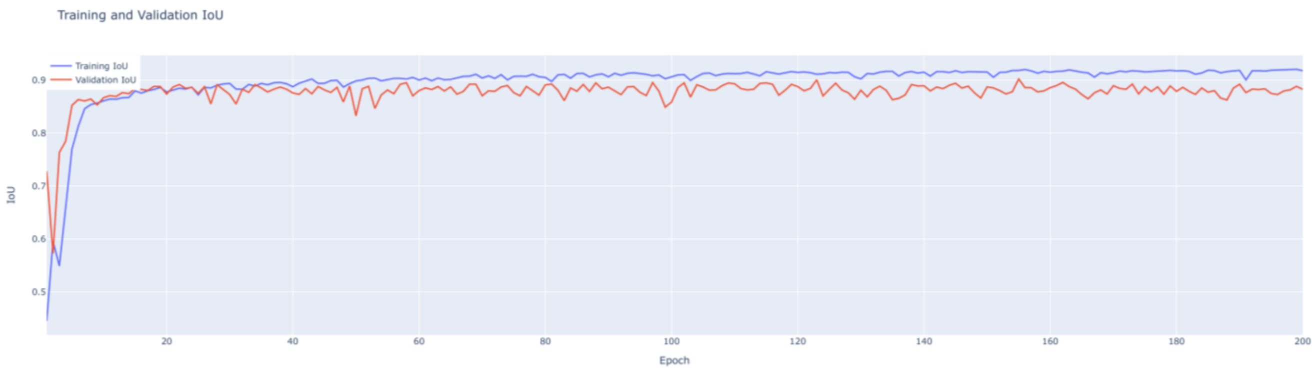


Figure 8

Plot of the training and validation Loss

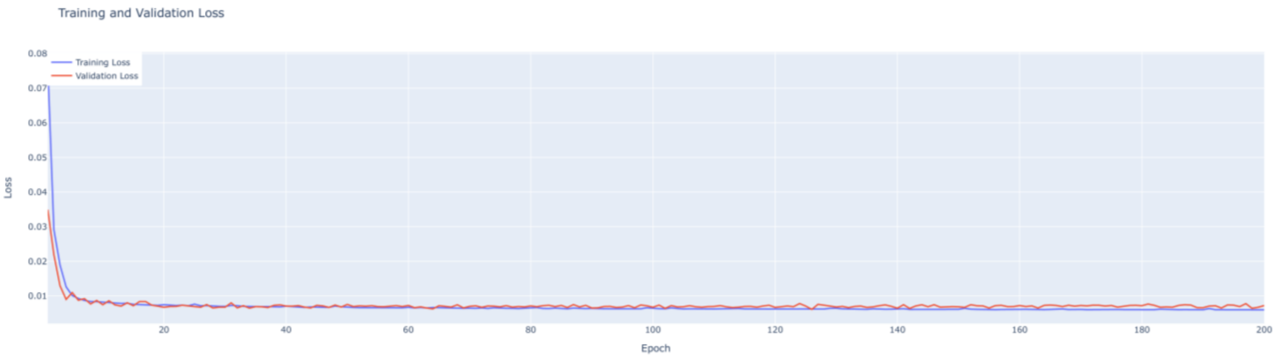


Figure 9

Model output example on the test set

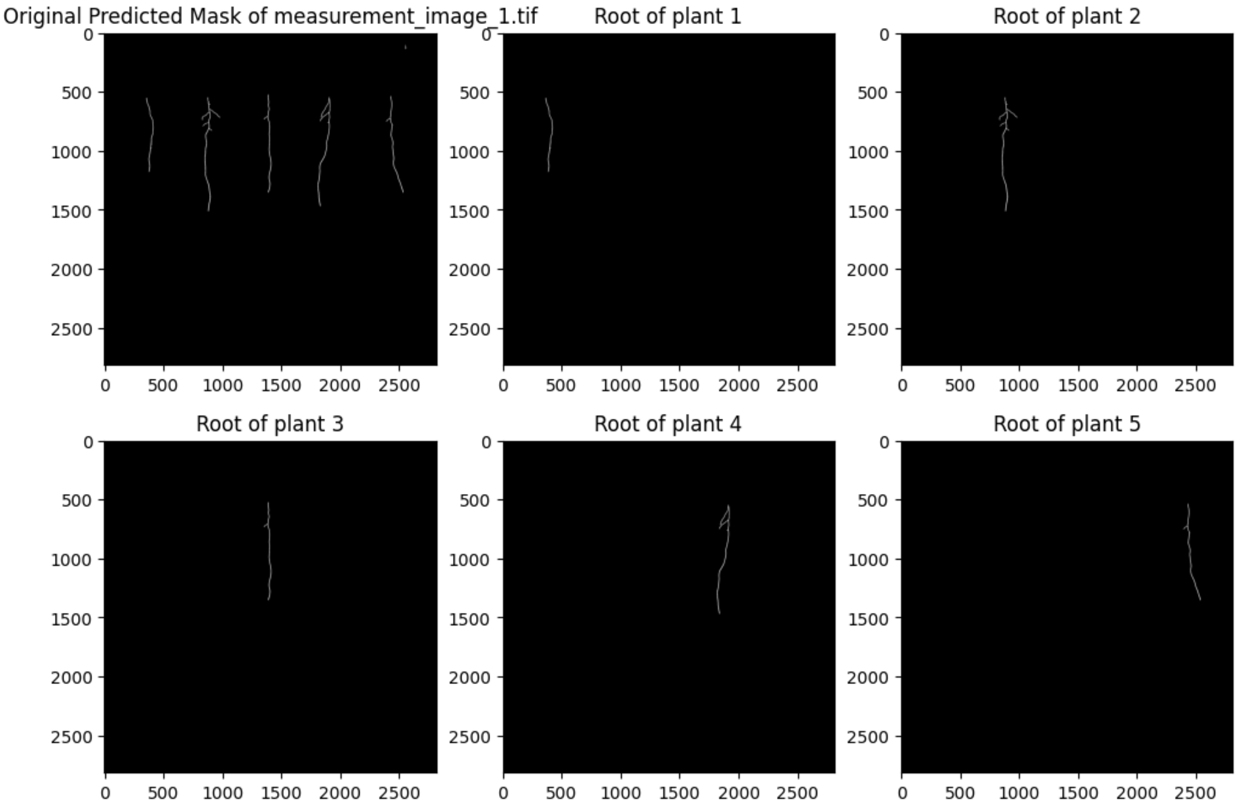
The solution attained a good performance but, not perfect. It captures certain water drops, scratches as noise and it does not capture occluded root (it wasn't trained to do so but this will have implications for the computer vision pipeline). Nonetheless, it respected all the client requirements.

1.4.5 Task 5 Instance Segmentation

This task involved again performing instance segmentation. However, this time our segmentation was completed not in one of the original dataset images but rather on the labels that the previously created root model created. My solution, like task 3 was able to meet all requirements, however in the CV pipeline it's possible to understand that in cases where the root plant was segmented as discontinuous (either due to poor performance or an occluded root) the segmentation process proves to be less precise and requires additional morphological operations such as dilation to reach a better precision.

Figure 10

Instance segmentation on the measurement dataset



1.4.6 Task 6 Landmark Detection

Landmark detection in computer vision refers to identifying specific points of interest within an image, often related to object or facial features.

In this project we used once again the output of our root model to find, (1) the primary root tip, (2) the junction between the root and the hypocotyl, (3) and the lateral root tips. To achieve these results, I employed skeleton analysis on my solution to correctly identify the different root landmarks.

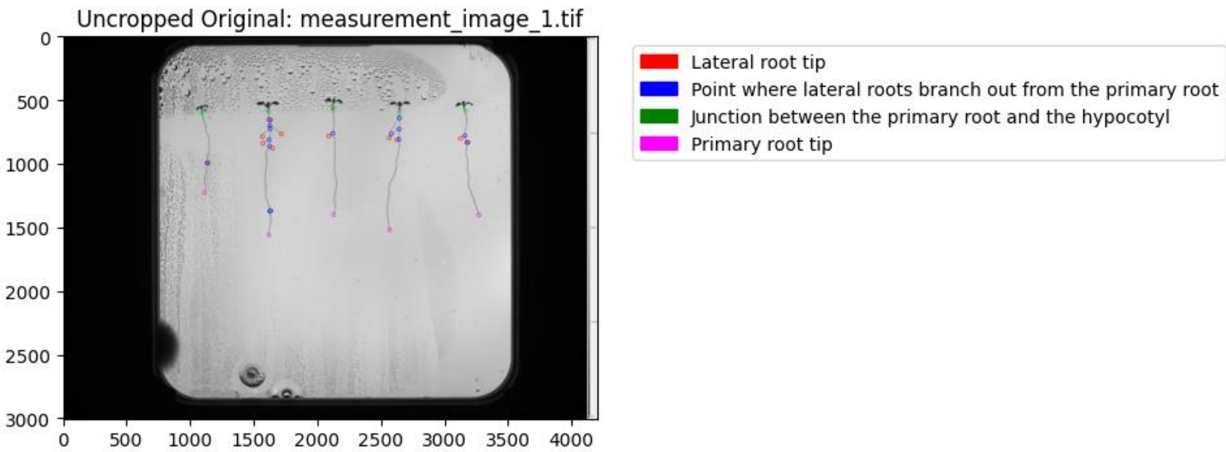
Figure 11

Example of root landmarks predictions.

My solution proved to perform within the client requirements of a maximum of a MAE (mean absolute error) of 100 pixels. On the measurement dataset, the overall MAE was between 3 and 11 pixels.

Figure 12

Example of Landmark detection solution



Performance metrics on the example above:

- MAE for junction between root and hypocotyl (x coordinate): 2.8
- MAE for junction between root and hypocotyl (y coordinate): 7.6
- MAE for primary root tip (x coordinate): 0.4
- MAE for primary root tip (y coordinate): 1.2
- Overall MAE: 3.0

1.4.7 Task 7 Morphometric Analysis

Morphometric analysis in computer vision refers to the quantitative measurement of shapes or structures within an image. It involves extracting and studying dimensions, such as lengths, areas, and volumes, to understand and compare object geometries.

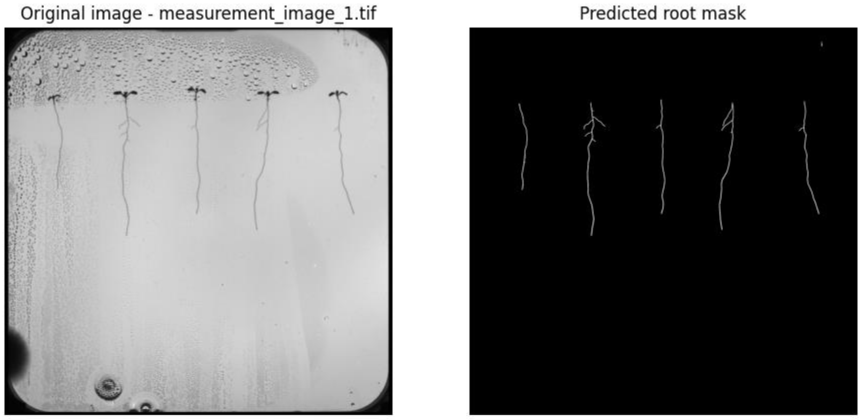
Building upon our work in task 6 we now performed morphometric analysis to calculate the primary root length and the total lateral root length in pixels.

The client requirements were that we attained a Symmetric mean absolute percentage error (SMAPE) of 20% or less for the primary root length and of 50% or less for the total lateral root length, in both cases, for the images of the Measurement Dataset.

Like task 6, my solution utilised skeleton analysis with the added calculation of root lengths with Dijkstra's algorithm, for finding the shortest paths between nodes in a graph.

Figure 13

Example of root output model



Here are the morphometric analysis results for the figure above:

Figure 14

Morphometric analysis results for the previous figure

Plant	Primary Root Length	Total Lateral Root Length
Plant 1	1100.546	852.703
Plant 2	1087.546	222.108
Plant 3	1037.747	223.108
Plant 4	938.250	93.497
Plant 5	1010.404	435.274

Overall Error Rates

- sMAPE for Primary Roots: **1.587%**
- sMAPE for Lateral Roots: **29.917%**

1.4.8 Task 8 The Kaggle Competition

For this task, we integrated the skills and knowledge acquired throughout Tasks 1 to 7, constructing a comprehensive computer vision pipeline. This pipeline will process raw images obtained from Hades and provide predictions for the primary root length of all five plants.

The students working on this project participated in a private competition hosted on the Kaggle platform to see which of us was able to obtain the best pipeline. The images used for the competition correspond to the Kaggle dataset mentioned previously.

For my solution I divided my pipeline into two jupyter notebooks:

1. 001_task_8.ipynb - data preparation

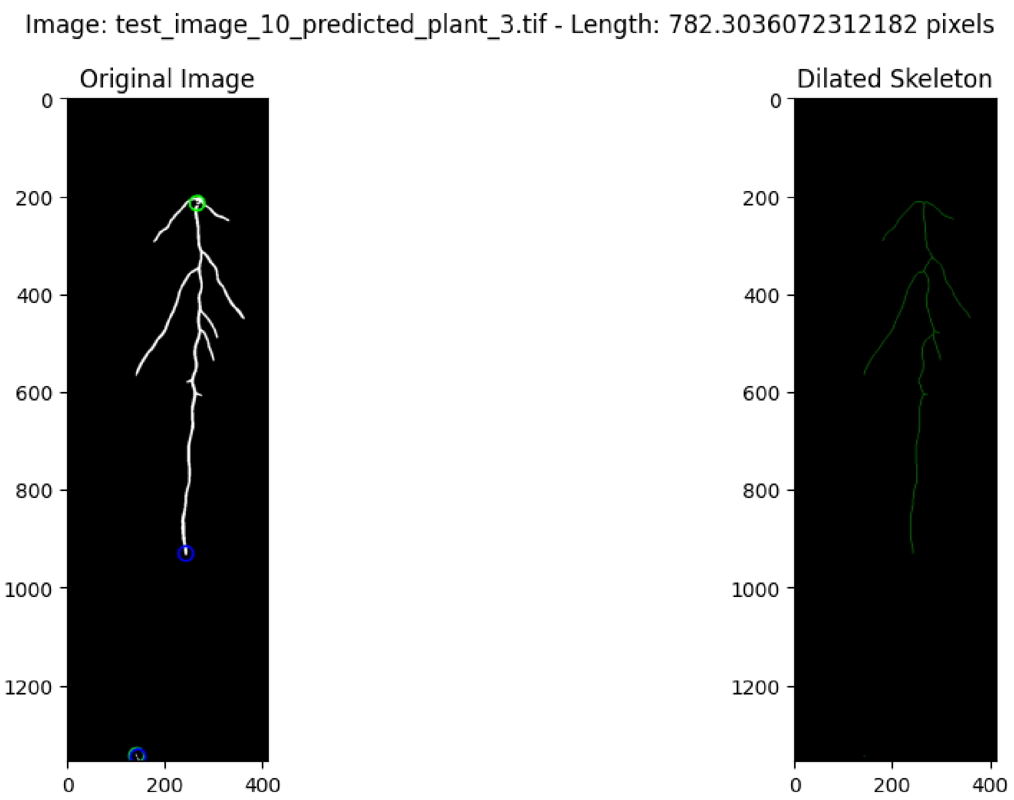
This notebook conveys the steps regarding the data preparation. Specifically, reading and cropping the images of the Kaggle dataset, applying the model's predictions to obtain the root mask, extracting and cropping each plant from each image and saving it to a specific folder.

2. 002_task_8.ipynb - Pipeline

This notebook conveys the steps regarding the CV pipeline. Specifically, reading each of the cropped plant's root, skeletonizing it, applying morphological operations, identifying the landmarks of primary root tip and junction between root and shoot and calculating the distance in pixels between the points and storing those values.

Figure 15

Example of the pipeline's calculation



I would also like to discuss the different experiments that led to my optimal solution. I performed the following experiments:

Applying different models which although trained with the same data, were:

- trained for a higher number of epochs 50 -> 100 -> 200
- a higher patience 11 -> 25 -> 55
- with horizontal data augmentation
- different batch sizes 32, 64, 128

Using different conditions for ROI to avoid picking up roots of seed which didn't germinate and plant roots in the wrong order:

- Adjusting the bounding box coordinates for the ROI:
- Removing noise, setting top and bottom areas of the images to a pixel value of 0

Different morphological operations:

- Closing
- Dilation
  - Use of different kernel sizes 3x3 and 5x5
  - Using different dilation iterations 2, 3, 4

Creating different processing functions for when the ROI has a single skeleton or multiple skeletons.

Combining these experiments, I reached a solution which obtained a SMAPE of 13.619, which is within the client's requirements.

The main constraint and design flaw of the pipeline is the assumption that the model's primary root prediction is continuous.

In cases for example, where the shoot grows on top of the root, the model is not able to predict the root which makes the root prediction appear to be broken into pieces. If the "break" is small, then morphological operations might be able to connect them again but in cases where the occlusion is bigger there is a need to find another strategy to perfect the root prediction, because even if you were able to sum to the pixel value the rest of the length of the primary root tip (which is a difficult task due to the skeleton of lateral roots) you would still be missing the pixel count of the occluded root.

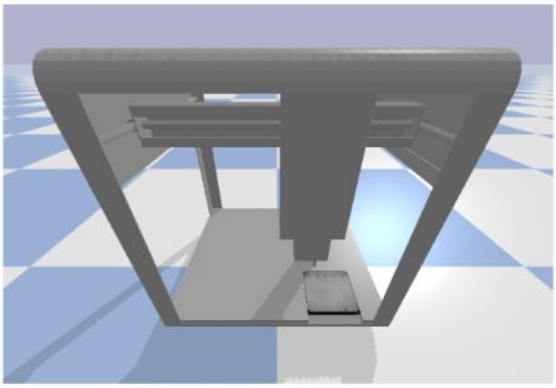
The model for the occluded root did not achieve effective results. I believe that a big motive for that is the labelling bias discussed in the dataset section. Nonetheless relabelling this class and attempting to build another model for it is a possible solution for this problem.

1.4.9 Task 9 Simulation Environment

For this task, we demonstrated our ability to give the robot commands and receive observations about the robot's state. We used the OT-2 Digital Twin which is a virtual representation of the Opentrons OT-2, a popular robotic liquid handling system used which is used at NPEC. My solution was able to manipulate the environment as expected.

Figure 16

Opentron OT-2 and the robotic simulation environment.



1.4.10 Task 10 Creating a Gym Environment

In this task we created a wrapper for the provided simulation environment mentioned in the last task to make it compatible with the gymnasium standard so that it can be used to train a reinforcement learning algorithm. The goal of the RL algorithm is to be able to move the tip of the pipette to any given position within the working envelope (the three-dimensional space that the robot can reach). For my wrapper solution I designed a rewards function that, for each step the robot takes, it evaluates the pipette position's distance to the goal and, compares the current distance to the previous one and gives a reward proportional to the improvement (regardless of it being a negative or positive improvement).

1.4.11 Task 11 Reinforcement Learning

For this task we trained a reinforcement learning algorithm using Stable Baselines 3 to control the Opentrons OT-2. The goal of the RL algorithm is to be able to move the tip of the pipette to any given position within the working envelope. For my task 11 solution, I used the Proximal Policy Optimization RL algorithm. I, alongside some of my colleagues Fedya, Martin, Kian, Benjamin, and Imani experimented with different reward functions from the task 10 wrapper as well as different hyperparameters for learning rate, steps and batch size. Each of us was in charge of testing certain parameters. I kept track of the different models I trained through the use of Weights and Biases (an experiment tracking tool). As a group, out of the parameters we tried we reached the conclusion that the best were:

- 128 for batch size
- 0.0003 for learning rate
- 2048 for number of steps

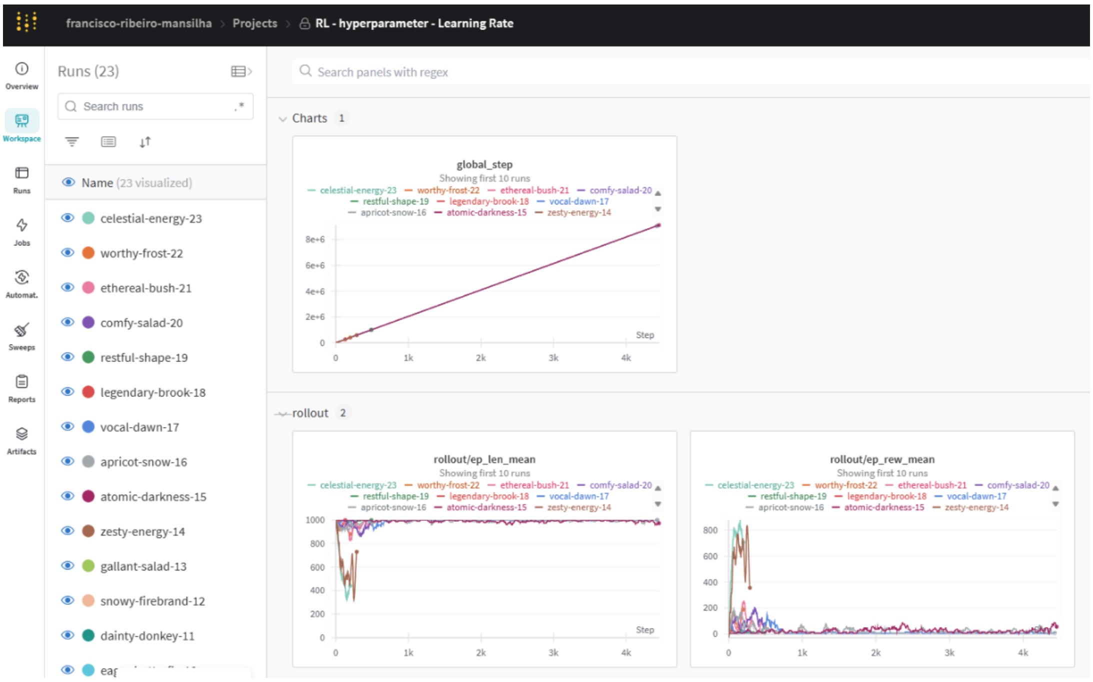
Figure 17

Table of the tested hyperparameters

Batch Size	Learning Rate	Number of Steps
------	0.01	------
64	0.001	512
$\boxed{128}$	0.0001	1024
256	$\boxed{0.0003}$	$\boxed{2048}$

Figure 18

Experiment tracking with weights and biases



1.4.12 Task 12 Creating a Controller

We also created a PID controller (a control algorithm) for the Opentrons OT-2. The goal of the PID controller is to be able to move the tip of the pipette to any given position within the working envelope. In my solution, I created 3 PID controllers one for each axis of the pipette position. The best performing gains for the PID controller are: proportional gain of 16, integral gains of 0 and a derivative gain of 0

Tasks 11 and 12 reach the same goal through different approaches.

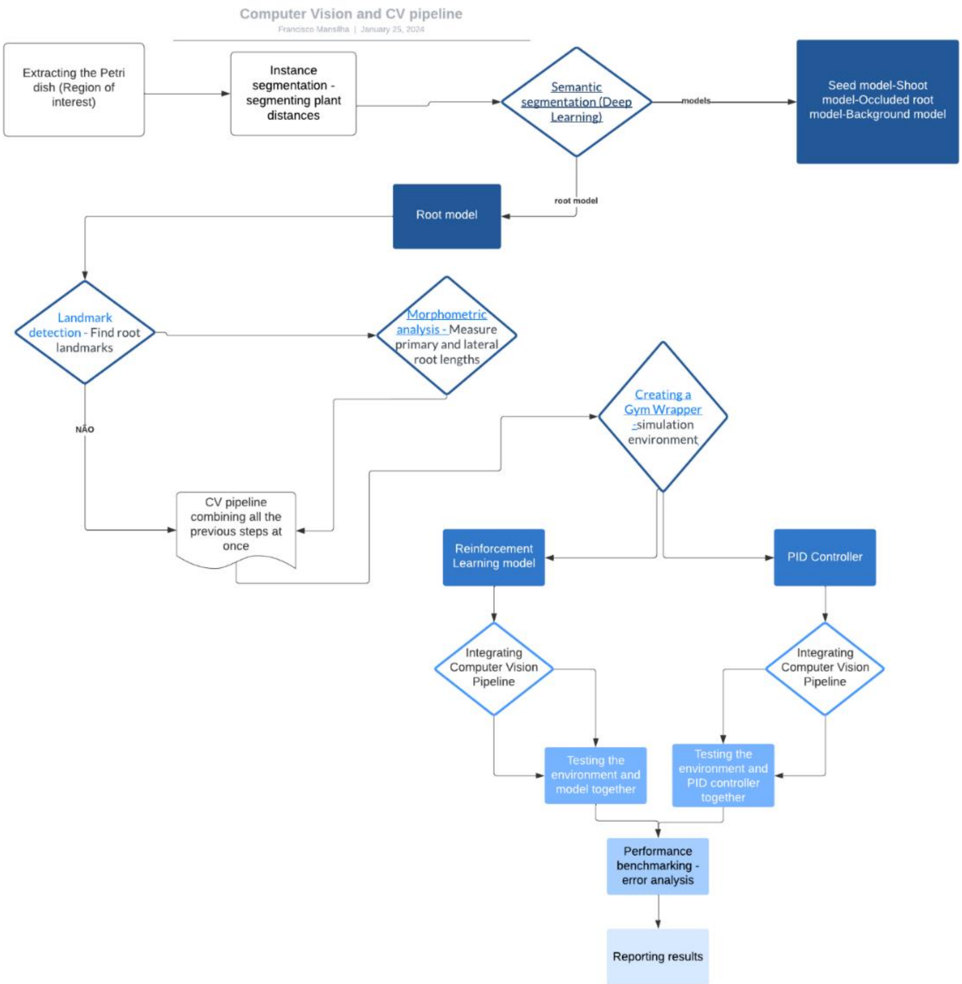
1.4.13 Task 13 Integrating the Computer Vision Pipeline

The main goal for this task is to inoculate the root tips present in multiple specimen plates. Therefore, after reaching a working computer vision pipeline, and a working controller, we integrate them. We modify the previously built controllers (PID controller and the RL controller) to accept the output of the computer vision pipeline as input.

To integrate the computer Vision pipeline into the RL and PID controllers all the previous datalab tasks mentioned previously, were utilized. This process can be summarized with the following flowchart:

Figure 19

Pipeline flowchart



There are steps about the pipeline integration which still weren’t mentioned, mainly the necessity to convert the landmark coordinates from pixel space into millimetre space and then into the robot’s coordinate space.

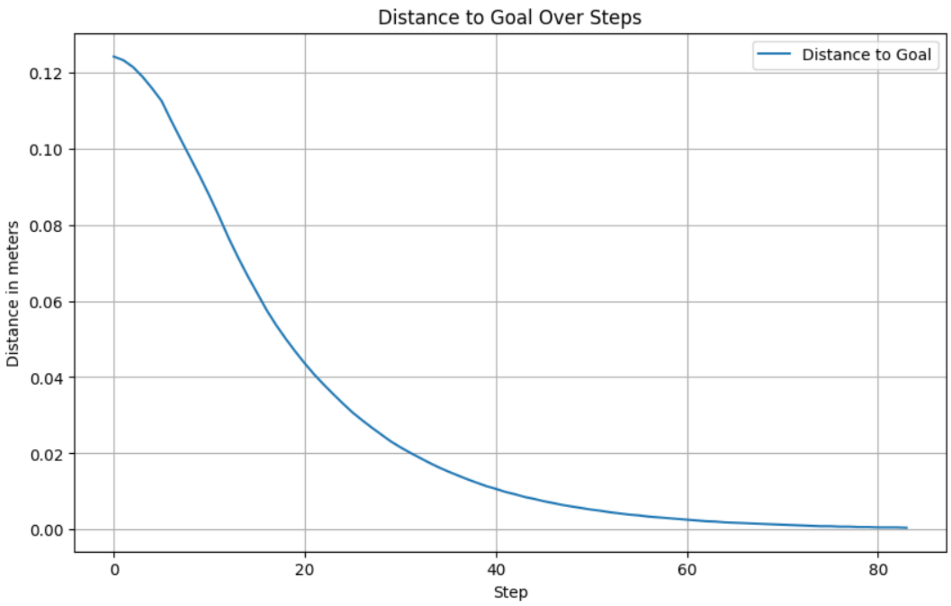
1.4.14 Task 14 Performance Benchmarking

This task corresponds to quantifying the performance of the systems we have created. We benchmarked the performance of the RL controller and the PID controller through different metrics, the most important ones being speed and accuracy.

In my solution, the PID controller performed much better than the RL controller both in terms of speed and accuracy. Whereas the RL controller reached an average error of 8 millimetres with an average speed of 1.12 seconds until inoculation, the RL controller reached an average error inferior to half a millimetre with an average speed of 0.8 seconds. Not only did I evaluate these metrics for both controllers I also created plots that quantify the distance to goal over the number of steps for each plant root.

Figure 20

Example of distance to goal plot



1.5 Results

The integration of computer vision, reinforcement learning, and robotics for the NPEC project successfully automated the plant root segmentation and inoculation process with notable precision and efficiency. The pipeline, designed to analyse Arabidopsis thaliana root systems, was meticulously evaluated through a series of tasks, demonstrating the project's innovative approach to plant phenotyping.

1.5.1 Pipeline Outcomes

Computer Vision Accuracy: The segmentation models, particularly the U-net architecture used for root detection, achieved a remarkable Intersection over Union (IoU) of 0.9029 on the test sub-dataset, surpassing the client's requirement of 0.5 IoU. Despite challenges such as distinguishing occluded roots and eliminating noise like water drops and scratches, modifications and rigorous validation ensured reliable segmentation (Figure 7).

Landmark Detection Precision: The precision in landmark detection within the plant roots was exceptionally high, with a Mean Absolute Error (MAE) ranging between 3 to 11 pixels for critical points in the measurement dataset. This precision underscores the effectiveness of the skeleton analysis technique used (Figure 9).

Morphometric Analysis: By employing Dijkstra's algorithm for root length measurements, the system maintained a Symmetric Mean Absolute Percentage Error (SMAPE) within the required thresholds, demonstrating the algorithm's adequacy for accurate morphometric analysis (Figures 10 and 11).

1.5.2 Integration Analysis

The project's success hinged on the seamless integration of individual tasks, where each component was pivotal to the overall functionality:

- Data Preparation and Segmentation laid the foundational accuracy for subsequent tasks by providing precisely segmented images.
- Landmark Detection and Morphometric Analysis utilized the segmented images to quantify root system characteristics, crucial for phenotypic analysis.
- Robotics Integration, particularly the PID controller, showcased superior speed and accuracy in inoculation tasks compared to the reinforcement learning approach, emphasizing the importance of choosing the right automation technique based on the application context (Figure 15).

1.5.3 Challenges and Solutions

Integration Complexity: The integration of disparate technologies—computer vision, deep learning, and robotics—posed significant challenges, particularly in maintaining data consistency and operational timing across systems. Solutions involved rigorous testing, iterative refinement, and synchronization mechanisms to ensure seamless workflow.

Data Bias and Noise: Encountered biases and noise within the dataset, such as capture bias and inconsistencies in labeling, necessitated the development of robust data preprocessing and augmentation techniques. These included manual corrections, enhanced labeling guidelines, and the introduction of noise-reduction algorithms to improve model training outcomes.

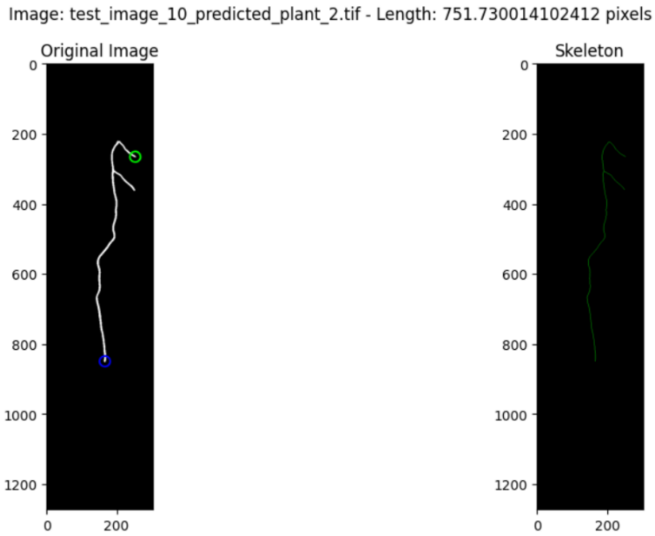
Robotics Precision: Achieving high precision in robotic inoculation was challenged by spatial discrepancies between computer vision outputs and the physical coordinate system of the robotic arm. This was resolved by implementing advanced calibration techniques and fine-tuning the controllers based on feedback loops through the simulation environment.

1.5.4 Possible Improvements

There is still margin for improvement on my CV pipeline. For example, I can improve the logic I use to find the junction point between the root and shoot since in a few rare cases that landmark wasn't properly identified:

Figure 21

Example of misidentified landmark



I can also improve if I create a strategy to deal with the occluded roots that I mentioned in task 8. I can try to perform different morphological operations or write an algorithm that would connect them for a more precise measurement.

Another area of improvement is the reinforcement learning controller. I can always try different hyperparameters and reward functions to obtain an error smaller than 1 milimeter.

1.6 Conclusion

This project represents a significant stride in the fusion of computer vision, deep learning, robotics, and biology to address the pressing needs of sustainable agriculture through the automated phenotyping of Arabidopsis thaliana. Our key findings include achieving an Intersection over Union (IoU) of 0.9029 for root segmentation, demonstrating precision in landmark detection with a Mean Absolute Error (MAE) as low as 3 pixels for some cases. The integration of these technologies facilitated the development of an automated inoculation process with a PID controller that outperformed its RL counterpart in both speed and accuracy, achieving an inoculation error of less than half a millimeter and an average speed of 0.8 seconds.

Reflecting on this journey, it's evident that the project was not just a technical challenge but a comprehensive learning experience. Engaging with complex datasets introduced me to the nuances of data bias and the importance of meticulous data preparation. The iterative process of model training, particularly with the U-net architecture and the Proximal Policy Optimization algorithm, honed my skills in deep learning and reinforcement learning, while the integration of computer vision outcomes with robotic controls underscored the criticality of precision in automation. The challenges encountered, from data noise to integration complexity, pushed me to delve deeper into the subjects, enhancing my knowledge in multiple areas. This project has significantly contributed to my professional growth, enriching my expertise in data science, deep learning, computer vision, robotics, and even biology.

In conclusion, the successful automation of plant root segmentation and inoculation for the NPEC project exemplifies the potential of integrating cutting-edge technologies to revolutionize plant science.

1.7 References

Angelique. (2021, March 16). Hades: high-throughput fluorescence-based phenotyping of in vitro root systems - NPEC. NPEC. https://www.npec.nl/news/hades-high-throughput-fluorescence-based-phenotyping-of-in-vitro-root-systems/

Noyan, M. A. (2022, June 9). Uncovering bias in the PlantVillage dataset. arXiv.org. https://arxiv.org/abs/2206.04374

Ronneberger, O. (2015, May 18). U-NET: Convolutional Networks for Biomedical Image Segmentation. arXiv.org. https://arxiv.org/abs/1505.04597

Unbiased Look at Dataset bias. (n.d.). https://people.csail.mit.edu/torralba/research/bias/

Games

Leisure & Events

Tourism

Media

Data Science & AI

Hotel

Logistics

Built Environment

Facility

Mgr. Hopmansstraat 2
4817 JS Breda

P.O. Box 3917
4800 DX Breda
The Netherlands

PHONE
+31 76 533 22 03
E-MAIL
communications@buas.nl
WEBSITE
www.BUas.nl

DISCOVER YOUR WORLD

Breda University
OF APPLIED SCIENCES