# *The use of Artificial Intelligence in plant phenotyping*

Kornelia A. Flizik

Breda University of Applied Sciences

*Data Science & Artificial Intelligence*

Block B Academic year 2023/2024

January 26, 2024

# 1. Introduction

## 1.1 Overview of the Project

In response to the critical role of plants in our future, this project aims to enhance plant phenotyping capabilities through the integration of Artificial Intelligence. Conducted in collaboration with the Netherlands Plant Eco-phenotyping Centre (NPEC) project focuses on the development and implementation of a computer vision algorithm connected to a liquid handling robot. The primary objective is to measure plant root length, identify specific root landmarks, and facilitate precise plant inoculation for efficient phenotyping. The project is scheduled to be completed within an 8-week timeframe.

With a strategic focus on establishing genotype-phenotype associations, the NPEC contributes significantly to the advancement of climate-resistant crops and sustainable cropping systems, ensuring the future's high-quality food production. The facility operates through six distinct modules, each dedicated to data collection in both macro and micro dimensions within controlled environments. The current project specifically targets the second module, Plant-Microbe Interaction Phenotyping, which incorporates advanced installations like Helios for plant shoots and Hades for in vitro root systems. The primary focus of this project aligns with Hades, which serves as a platform for automated processing of seedlings on Petri dishes. Hades provides features such as automated filling, sowing, vernalization, and controlled environmental conditions. Notably, it enables the automatic treatment of roots with microbes or synthetic communities, contributing significantly to the project's objectives.

## 1.2 Background

### Plant Science

Plant phenotyping is the process of studying and measuring plants' observable characteristics (phenotypes) that result from their genetics' interaction with the environment. For example, the height of a corn plant is a phenotype that can be influenced by both its genetic makeup and environmental factors such as soil quality and sunlight. Plant phenotyping includes methodologies and protocols that are used to measure plant growth, architecture, and composition.

In the evolving landscape of plant science, advancements in phenotyping technologies, particularly those employing computer vision and machine learning, have paved the way for non-invasive and high-throughput analyses.

### Use of Computer Vision, Reinforcement Learning and Robotics

A possible solution integrates computer vision, reinforcement learning and robotics to improve plant phenotyping. A computer vision algorithm can process image data, extract and segment roots, and identify key landmarks. A robotics system based on reinforcement learning uses these landmarks to identify optimal inoculation locations and perform precise and effective interventions. This combination of technologies aims to provide a comprehensive solution for high-throughput plant phenotyping.

# 2. Data

## 2.1 Data Source

I have gained access to the data through the Breda University of Applied Sciences GitHub classroom page. The dataset provided by Hades comprises monochromatic morphometric images featuring Arabidopsis thaliana. These images depict the growth stages of the plants, which were cultivated in Petri dishes set against a black background. Each image in the dataset is sized at 4202 x 3006 pixels.
In the experimental setup, five seeds are initially planted in each Petri dish and documented through daily photography. Over time, the seeds sprout, leading to the appearance of roots and shoots. The growth process is visually depicted in Figure 1.

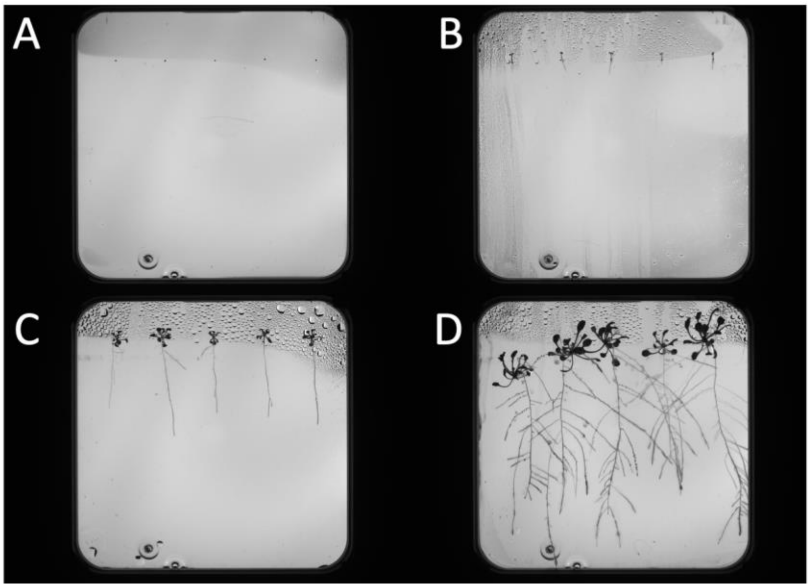


*Figure 1: The growth stages of Arabidopsis thaliana in Petri Dish.*

The data has been divided into four distinct datasets, including training data, test data, a Kaggle dataset, and a measurements dataset (as outlined below).

| **Dataset:** | **Number of images:** | **Format:** |
|---|---|---|
| train | 96 images | .png |
| test | 30 images | .png |
| Kaggle | 11 images | .tif |
| measurements | 3 images (2 used) | .tif |

The data labels are represented as masks in the .tif format, with each image associated with four distinct labels: shoot, seed, roots, and occluded roots. These masks serve as binary images with pixel values of 0 and 1, maintaining the same shape as the original images. The inclusion of occluded root labels is particularly crucial for the model to accurately identify and differentiate between various elements in the images. This addition allows the model to predict root lengths while accounting for instances where roots may be partially obscured or hidden.

## 2.2. Data quality

To identify a potential data bias, it is crucial to understand project objectives and goals. The goal of the computer vision solution is to enhance plant phenotyping processes in the lab environment. As the environment is structured, the data does not need to represent different environments and use-cases. However, it is still crucial to identify data biases, quality and limitations for future improvements and wider implementation. When dealing with image data of plants in Petri dishes captured by an Opentrons OT-2 robot, there can be several potential biases and challenges that may affect the quality and representativeness of the data.
Here are some potential biases to consider:

### Dataset Collection:

1. Single Camera Perspective: The unidirectional nature of the dataset, acquired through a single camera situated above the Petri dish, introduces a potential sampling bias towards a top-down view. This limitation may compromise the model's capacity to generalize across various root orientations.
2. Petri Dish Size and Shape: The size and shape of the Petri dish may affect the perceived size and shape of the plants within. The represented Petri Dish is square; however, this could differ across labs.

### Potential Biases:

1. Misinterpretation of Artifacts: The dataset is susceptible to misinterpretation, where the model might erroneously identify non-biological artifacts (e.g., scratches) as roots. This source of bias necessitates a meticulous pre-processing step for artifact removal.
2. Perspective Bias: The model may exhibit suboptimal performance on roots with intricate three-dimensional structures due to a predominant top-down perspective. Addressing this bias requires augmentation techniques that simulate diverse viewing angles.

### Limitations:

1. Limited Species Representation: The model's restricted training solely on Arabidopsis roots poses a generalization challenge when applied to roots from different plant species (e.g. potato).
2. Growth Stage Variation: Inadequate representation of some growth stages within the training dataset (lack of early growth stage) hampers the model's ability to generalize across different developmental phases.

# 3. Project process

## 3.1. Image annotation

Image annotation in computer vision is a process that involves marking specific regions or features within an image. In the context of this project, data labels are represented by four distinct binary images: shoot, seed, roots, and occluded roots. To achieve those masks, image annotation needed to be conducted on a dataset. Due to the academic nature of the project, the image labeling process was distributed among multiple participants, with each individual responsible for labeling 2-3 images. This collaborative effort spanned a period of 9 days, ensuring a thorough and meticulous labeling process. Additionally, every mask underwent a quality check to meet the desired standards and conditions, contributing to the overall reliability of the annotated dataset.  The correct masks example is presented in Figure 2.

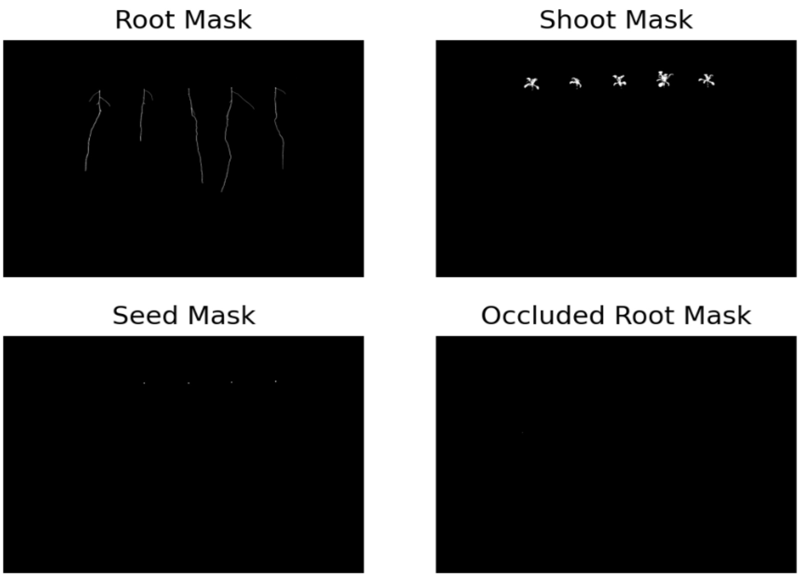


*Figure 2: Example masks results after image annotation.*

## 3.2.  Petri dish extraction

In the image preprocessing pipeline, a critical step involved the extraction of the Region of Interest (ROI), specifically the Petri Dish, from the black background. This extraction process is essential for enhancing the performance and efficiency of subsequent computer vision tasks. Traditional computer vision techniques were employed in this step. This non-machine learning approach was chosen for its suitability in handling preprocessing tasks, where simplicity and efficiency are crucial.
The steps employed in this task include thresholding, contour Identification and finding bounding box coordinates. The output is a cropped, square image showing Petri Dish. This result can be seen in Figure 3.

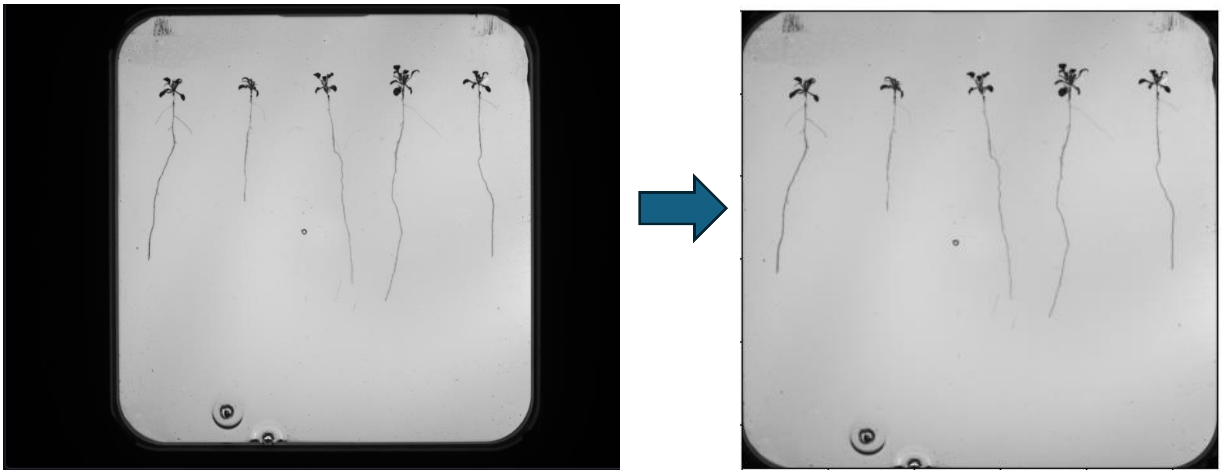

*Figure 3: Original and cropped image from train dataset.*

## 3.3. Instance segmentation

Instance segmentation is a task that involves classifying and precisely delineating individual objects, or instances, within an image at the pixel level. In contrast to semantic segmentation, which groups objects of the same class together, instance segmentation goes a step further by distinguishing each unique object.
In the specific context of our project, instance segmentation was applied to effectively segment each plant instance present in the images. Notably, the system is designed to handle any number of plants, up to a maximum of 5 instances.
To tackle this challenging task, I leveraged traditional computer vision techniques. This approach, which does not rely on machine learning models, proved effective in accurately classifying and delineating individual plant instances, contributing to the overall success of our image analysis pipeline.
In Figure 4, instances of plants are visually highlighted and distinctly marked with different colors. Each color corresponds to a unique plant instance, providing a clear and intuitive representation of the individual entities within the image.

*Figure 4: Entire plant instance segmentation.*

## 3.4. Semantic segmentation

In the subsequent phase of the computer vision pipeline, semantic segmentation took place to discern different plant organs instead of individual plants. This involved training a machine learning model, specifically a deep learning model, on raw images, treating each plant organ as a distinct category.

### 3.4.1. Data

It is worth noting that the training dataset was relatively small, consisting of only 96 images. Consequently, several preprocessing steps were implemented to enhance model performance:

1. Cropping and Padding: Raw images underwent initial cropping to eliminate black regions, optimizing the subsequent training process. Following cropping, each image was padded to ensure uniform patches. The resulting padded images had a size of 2816 x 2816 pixels, involving the addition of 26 pixels on each side around the Petri Dish.
2. Image Patching: Padded images were then segmented into patches of the desired size, with a selected patch size of 256 pixels x 256 pixels. This specific choice aimed to maintain an unbiased representation of the data while supplying a sufficient volume of data for effective model training. Notably, the total number of patches generated was 121, equivalent to 11 images along the length of the Petri Dish.

These preprocessing steps were methodically applied to both the training and test sets, aligning with the corresponding masks. The resulting dataset, which included segmented and patched images, served as the foundation for training the machine learning model to discern between different plant organs in the subsequent stages of the computer vision pipeline.
The organized data patches were stored in separate folders, making it easy to access and seamlessly integrate into the training pipeline.
The dataset included 9.559 images, representing patches extracted from the original images, alongside 9.559 images for each of the four mask types.
To ensure robust performance, the data was normalized, and a 4:1 split was applied to create training and validation sets.

### 3.4.2. Models

To accomplish the task objective, four distinct binary models were employed, each dedicated to a specific mask type. This deliberate approach ensured the creation of separate models for different plant organ categories, with the goal of achieving more refined and specialized segmentation results, particularly for roots. This modular model design sets the foundation for continued progress in the project.
The architecture of each model utilized the U-Net structure, featuring a contraction path and an expansive path. The model's output consisted of a sigmoid convolutional layer with a shape of (1,1).
Each binary model was trained on 7648 images. The training process incorporated early stopping, with a focus on tracking the best Intersection over Union (IoU), a pivotal metric for assessing the model's accuracy.

### 3.4.3. Models training and results

The results of the models were assessed on Test dataset.

| Model | Training time | IoU performance |
|---|---|---|
| Root model | 27 epochs | 0.9013196 |
| Shoot model | 30 epochs | 0.93527216 |
| Seed model | 15 epochs | 0.96715367 |
| Occluded root model | 7 epochs | 0.98278236 |

The model's IoU results are high, meeting the client's requirements. However, it is crucial to note that the final score may be affected by the abundance of blank pixels in the images. To improve score representability, further steps can be taken in subsequent analysis stages. This involves refining the scoring methodology to better consider the influence of blank pixels, ensuring a more detailed and comprehensive evaluation of the model's performance.

#### Root Model

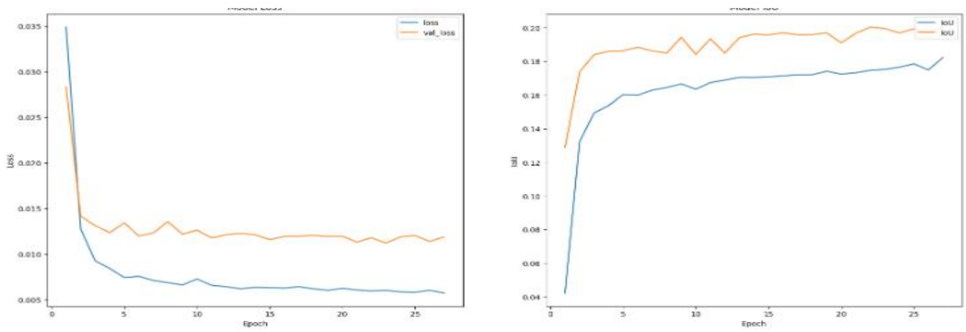

*Figure 5: The metrics score (loss and IoU) of training and validation sets.*

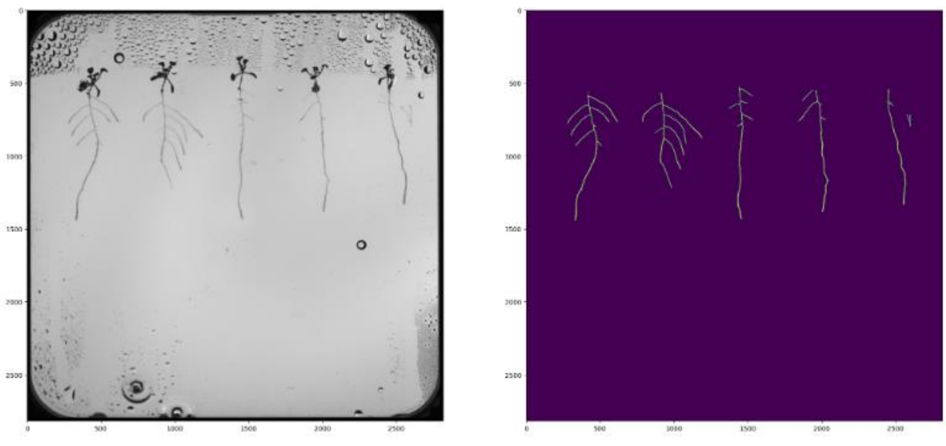

*Figure 6: Root model's predictions.*

## Shoot model

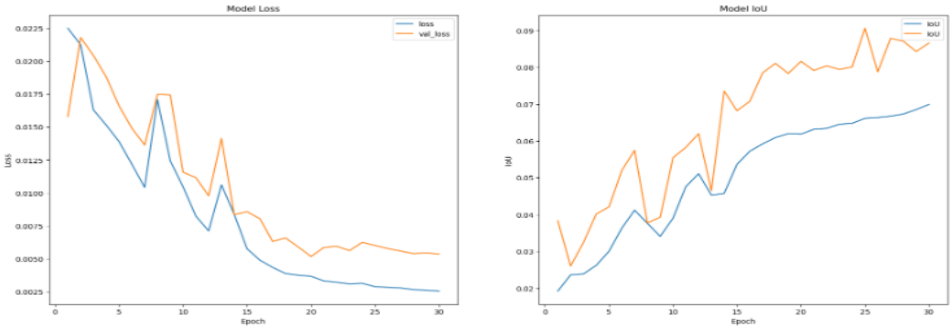


*Figure 7: The metrics score (loss and IoU) of training and validation sets.*

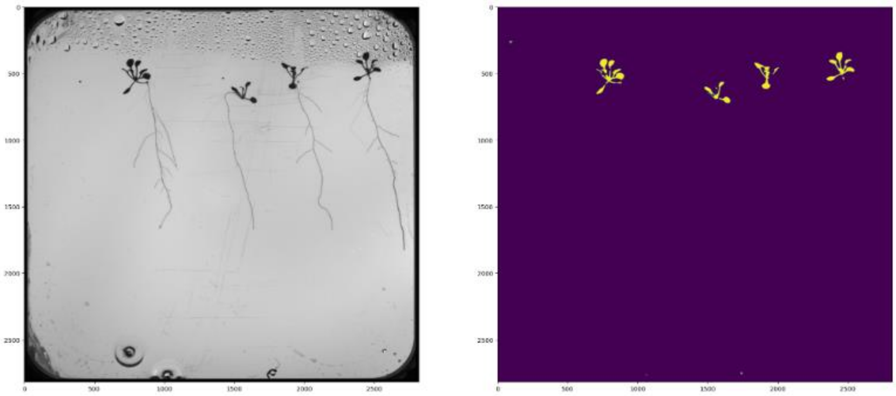

*Figure 8: Shoot model's predictions.*

## Seed model

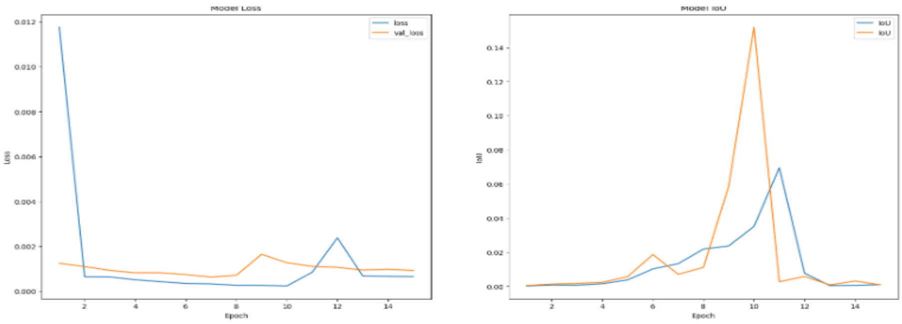


*Figure 9: The metrics score (loss and IoU) of training and validation sets.*

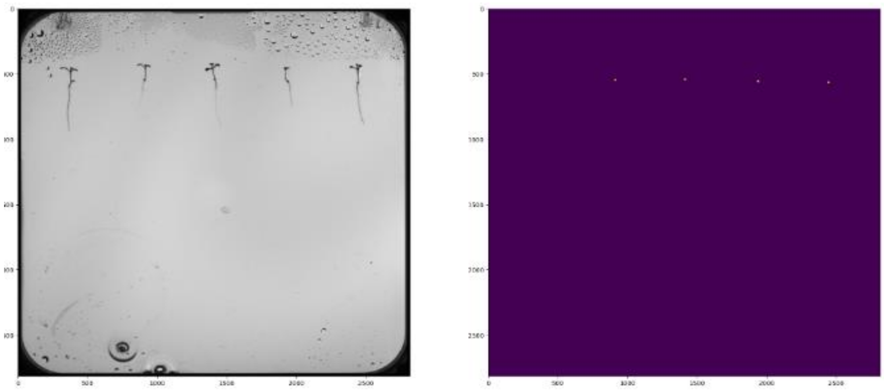
,

*Figure 10: Seed model's predictions.*

## 3.5. Instance segmentation

Following the effective model training outlined in Section 3.4, instance segmentation was implemented to identify individual root instances within the images. In this phase, the previously trained root model was employed to predict roots in a sample image extracted from the Measurement dataset. The segmentation of instances was executed using a methodology akin to the solution described in Section 3.3.

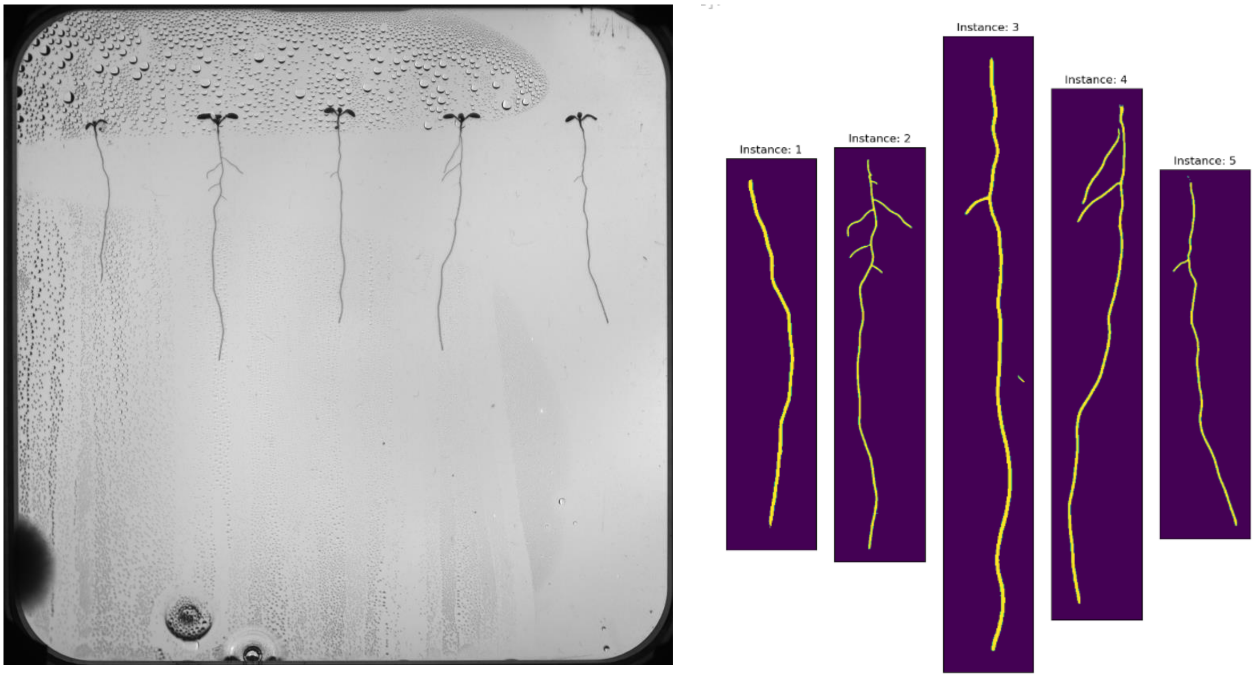


*Figure 11: Root instance segmentation.*

## 3.6. Root landmark detection

Landmark detection in computer vision is the process of identifying specific points of interest within an image, and in this project, these points correspond to landmarks on the plant's roots.

To tackle this task, a skeleton analysis was conducted, using techniques like skeletonization and graph representation. The outcomes of this analysis are visually presented in Figure 12, and the mean absolute error (MAE) score is provided for quantitative evaluation. The results were plotted and evaluated on a Measurement dataset. The MAE was calculated by measuring the discrepancy between the actual junction point between the primary root and the hypocotyl (marked in green), the primary root tip (marked in black), and their predicted coordinates. The MAE score is recorded at 73.35, aligning with the client's specific requirements.

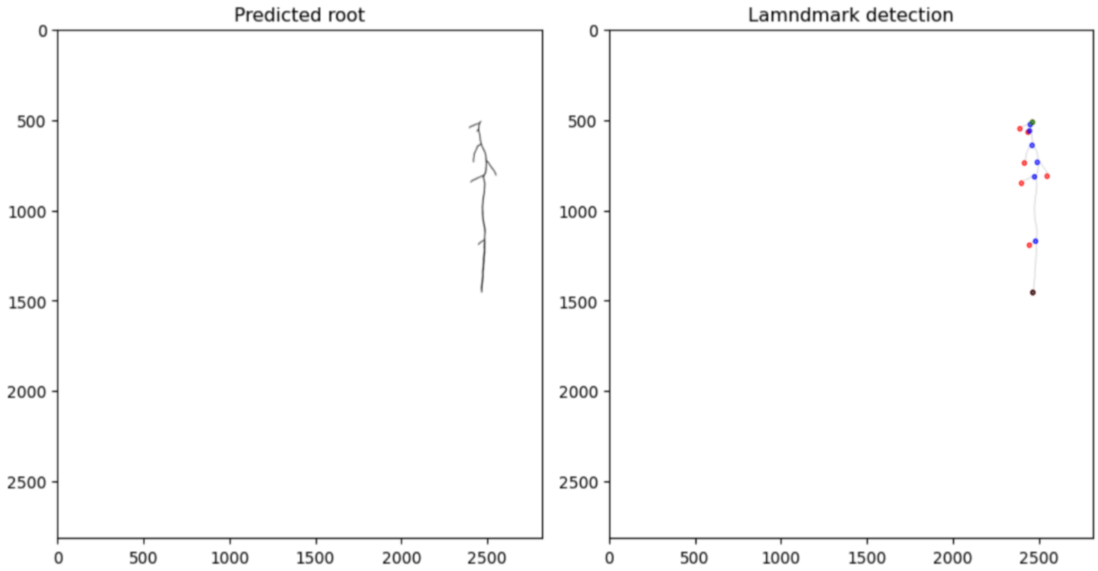


*Figure 12: Root landmarks. The black circle is the primary root tip. The green circle is the primary root start. The red circles are the lateral root tips. The blue circles are the landmarks where lateral roots branch out from the primary root.*

## 3.7. Morphometric analysis

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

In the context of this project, morphometric analysis serves as the concluding step in the computer vision pipeline. For each plant instance, precise measurements were obtained for the primary root length and total lateral root lengths. These results were then evaluated using the Measurement dataset.

The performance assessment utilized the Symmetric Mean Absolute Percentage Error (sMAPE) score. The formula below represents how the sMAPE score is calculated:

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

*where $A_t$ is the actual value and $F_t$ is the forecast value.*

The outcomes for primary root length and total lateral root lengths are 4.95% and 8.7%, respectively.

## 3.8. Simulation Environment

The OT-2 Digital Twin stands as a virtual representation of the Opentrons OT-2, a widely used robotic liquid handling system prevalent in laboratory settings. This digital twin is established within a simulation environment crafted with the PyBullet physics engine. Key features of the simulation environment include multiple robotic entities arranged in a 3D grid pattern, each equipped with specimens. These robots mimic real-world actions, facilitated by simulated pipettes, enabling the controlled dispensing of

droplets onto the attached specimens. The environment is designed to provide comprehensive control over the robots' movements, allowing for precise interactions with the specimens.
Functionality within the simulation encompasses initialization, resetting, running, applying actions, and retrieving states. The working envelope of the robot's pipette movements is defined as follows:

| **Axis** | **min** | **max** |
|---|---|---|
| x | -0.187 | 0.253 |
| y | -0.1709 | 0.2195 |
| z | 0.1195 | 0.2895 |

To better understand the robotics objectives, Figure 13 shows the Opentrons OT-2 and its digital representation OT-2 Digital Twin.

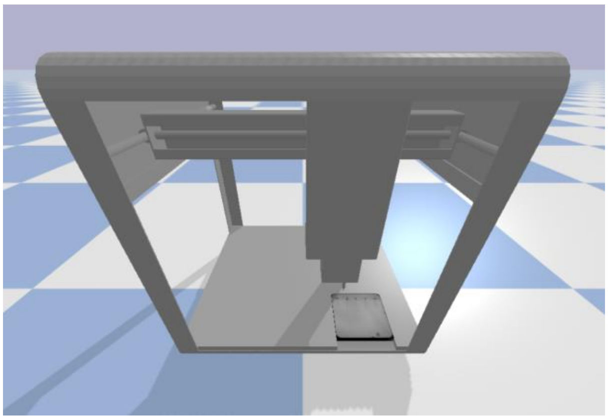

*Figure 13: Opentrons OT-2 and its digital representation in the environment*

## 3.9. Creating a Gym Environment

A crucial step in the robotics pipeline involves the creation of a Gym wrapper for the provided PyBullet simulation environment, ensuring compatibility with the Gymnasium standard. This adaptation is essential for seamlessly integrating the simulation into the reinforcement learning (RL) workflow. To empower precise pipette movements within the defined working envelope, essential properties and methods were implemented in this dedicated wrapper.

### Properties

1. Action Space: Defines possible movements within the working envelope.
2. Observation Space: Provides information about the current state of the environment.
3. Possible Actions: Specifies permissible movements and instructions that the agent can provide to the simulated robotic system.

4. Possible Observations: Describes the available information for the agent regarding the current state of the environment.
5. Reward Function: Quantifies the agent's performance, assigning a numeric value based on the desirability of its actions.
6. Done Condition/s: Determines when an episode is considered complete.

### Methods

1. Reset: Initializes or resets the simulation environment to its initial state.
2. Step: Executes actions, transitions the environment to a new state, and provides associated feedback crucial for learning.
3. Render: Visualizes the environment's current state.
4. Close: Terminates the simulation environment.

The wrapper was tested by running the environment for 1000 steps with random actions. Conducted tests were successful.

## 3.10. Creating a Controller

Proportional-Integral-Derivative (PID) controllers play a significant role in regulating various industrial process variables in today's automatic process control applications. The classic PID controller is widely used in the industry due to its simplicity and user-friendly nature.

To precisely position the tip of the pipette at any desired location within the working envelope, three PID controllers were developed, each dedicated to a specific axis. These controllers were integrated with the Opentrons OT-2 wrapper.

Understanding the function of PID gains—Proportional (Kp), Integral (Ki), and Derivative (Kd)—is essential for effective control system design. Each gain serves a specific purpose:

1. Proportional Gain (Kp) is a measure of system stiffness. It determines the amount of restoring force that should be applied to overcome position errors.
2. Integral Gain (Ki) is related to static torque load on the system. The Ki value “pushes” the system to zero positioning error at the end of the move.
3. Derivative Gain (Kd) represents the damping effects on the system, working with proportional gain to reduce overshot and oscillations.

The process of determining the optimal gains involved subjecting the controller to twenty iterations, each consisting of one hundred steps. Following each iteration, the gain values were adjusted. The graphical representation of the best gains finding process is illustrated in Figure 14. Best gains:

| **PID controller** | **Kp** | **Ki** | **Kd** |
|---|---|---|---|
| x_pid | 11 | 0.0 | 3 |
| y_pid | 16.5 | 0.0 | 3 |
| z_pid | 16.5 | 0.0 | 3 |

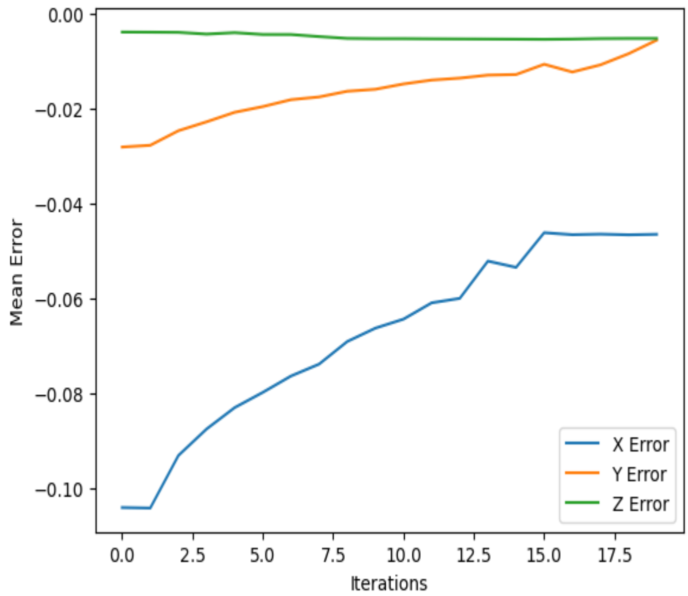

*Figure 14: Finding the best gains for PID controllers.*

# 4. Overall results

## 4.1. Project progress

### Computer vision

The computer vision pipeline is able to take raw images from Hades and compute primary root length using root landmarks (start and tip of the root).
Summary of the pipeline:

1. Data Labeling (Section 3.1): The labeling process was crucial for both model training and evaluation, forming the foundation for the next pipeline stages.
2. Image Cropping (Section 3.2): The accurate cropping of images significantly contributed to the high precision of the model, ensuring focused analysis on relevant regions.
3. Semantic Segmentation (Section 3.4): The use of a model to identify plant organs. This model allowed isolating roots in the image and was crucial for plant phenotyping and main objectives of the project.
4. Instance Segmentation (Sections 3.3 and 3.5): This step enabled the system to identify individual plants and execute actions independently for each.
5. Landmark Detection (Section 3.6) and Root Measurements (Section 3.7): The solution achieved in these stages met the client's requirements, delivering successful landmark detection and precise root measurements.

The final pipeline was evaluated using a data science competition platform called Kaggle, using the dataset described in Section 2.1. While image data was accessible to all project contributors, the output was concealed to prevent adjustments that might artificially lower errors. The client's requirement for this challenge was to achieve an error of sMAPE of less than 50%. The achieved result is 23.077% meeting the initial goal. The dataset uploaded to Kaggle was direct output from my pipeline, with two columns representing image name and primary root length.

The outcomes of the Kaggle competition underscore the success of the proposed solution, affirming its effectiveness in achieving the specified objectives.

### Robotics

The robotics pipeline provides a solid foundation for the integration of a functional computer vision pipeline and an effective controller, ensuring accurate pipette positioning for the inoculation of the root system.
Summary of the pipeline:

1. Environment testing (Section 3.8): The initial step involved learning and testing the environmental properties to establish cooperation with the agent. This step was crucial for further robotics tasks.
2. Gym wrapper (Section 3.9): Wrapper with essential properties and objectives was crucial for seamlessly integrating the simulation into the reinforcement learning workflow, making it compatible with the Gymnasium standard.
3. Controllers (Section 3.10): Creating PID controllers is crucial for precise pipette movements. This allows the agent to inoculate specified locations (landmarks).

Due to time constraints, the last steps including integration with computer vision and performance benchmarking were not finished. However, the resulting pipeline still offers a solution with an open space for improvements and further actions.

# 5. Conclusion

## 5.1. Limitations

A significant constraint in this project was the limited timeframe. The project's objectives required the implementation of various solutions, particularly the time-intensive process of training models, causing a notable delay in the overall workflow. Notably, the optimization efforts in improving the model's performance in Section 3.4 extended the duration of the computer vision pipeline beyond the initial estimates. Consequently, the planned reinforcement learning, and integration phase could not be completed within the designated timeframe.

## 5.2 Recommendations

Moving forward, it is recommended to implement a landmark detection model to guide the pipette to specified coordinates effectively. A pivotal next step involves testing both Reinforcement Learning (RL) and Proportional-Integral-Derivative (PID) controllers to determine their respective performances.
Upon the successful implementation of the robotic system, a comprehensive evaluation is advised through performance benchmarking. This evaluation should focus on identifying the best-performing controller, with a specific emphasis on assessing speed and accuracy. These key metrics will provide valuable insights into the overall efficiency and reliability of the developed robotic system.

## 5.3. Key Findings

The project has made significant progress in both computer vision and robotics pipelines, showcasing achievements in primary root length measurement through image analysis and the development of a foundation for precise pipette positioning in the inoculation process. The computer vision pipeline successfully implemented data labeling, image cropping, semantic and instance segmentation, as well as landmark detection and root measurements. The achieved sMAPE error of 23.077% in the Kaggle competition confirms the effectiveness of the proposed solution.

In robotics, initial steps involving environment testing, Gym wrapper creation, and PID controller implementation were completed. Although time constraints prevented the integration with computer vision and performance benchmarking, the developed pipeline sets the stage for further enhancements.

## 5.4. Learning Objectives

The collaboration with NPEC provided a valuable learning experience and skill development opportunity. The academic nature of the project facilitated the acquisition of knowledge and skills in diverse areas such as computer vision, machine learning, reinforcement learning, and robotics. The practical application of these concepts in a real-world scenario, coupled with the challenge of meeting client requirements, enhanced the understanding of the complexities involved in AI projects. The project served as a valuable learning experience for the role of an Artificial Intelligence Scientist.

# References


1. Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., & Bennett, M. (2017). Plant phenomics, from sensors to knowledge. Current Biology, 27(15), R770-R783.

2. Subhedar, J., Bachute, M., Koundal, D., & Kotecha, K. (2023). Semantic Segmentation Algorithm for Autonomous Driving using UNET Architectures: A Comparative Study.

3. A. Torralba and A. A. Efros, "Unbiased look at dataset bias," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 1521-1528

4. Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. PloS one, 12(3), e0174202. https://doi.org/10.1371/journal.pone.0174202

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

6. Breda University of Applied Sciences. (n.d.). DataLab Tasks. GitHub. https://adsai.buas.nl/Year2/BlockB/DataLab%20Tasks.html

7. Li, Z., Guo, R., Li, M., Chen, Y., & Li, G. (2020). A review of computer vision technologies for plant phenotyping. Computers and Electronics in Agriculture, 176, 105672. https://doi.org/10.1016/j.compag.2020.105672

8. Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., & Melgani, F. (2018). Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. GigaScience, 8(1). https://doi.org/10.1093/gigascience/giy153

9. Collins, D. (2022, October 17). FAQ: What are PID gains and feed-forward gains? Motion Control Tips. https://www.motioncontroltips.com/faq-what-are-pid-gains-and-feed-forward-gains/#:~:text=Proportional%2C%20integral%2C%20and%20derivative%20gains,a%20measure%20of%20system%20stiffness.