Technical Report

Dominik Szewczyk
224180
Academic Year: 2023-24 Y2B
Instructor: Dean van Aswegen Jason Harty Alican Noyan
Client: Netherlands Plant Eco-phenotyping Centre (NPEC)
Submission Date: Thursday 25th January, 2024

1 Introduction

1.1 Overview

The Netherlands Plant Eco-phenotyping Centre (NPEC) has presented us with an intriguing challenge, seeking a solution that involves segmenting plant roots from images and controlling a liquid handling robot for precise inoculation of plants. Before delving into the technical intricacies, it's crucial to grasp the fundamentals of plant biology to successfully address this project.

1.2 Importance of Plant Science

Plants play a pivotal role in our sustenance, providing direct and indirect sources of food. NPEC endeavors to leverage plant potential for future food and material needs, emphasizing sustainability through cutting-edge plant phenotyping facilities.

1.3 Plant Phenotyping and Its Modules

Plant phenotyping involves studying and measuring observable plant characteristics resulting from the interaction of genetics with the environment. NPEC, equipped with seven specialized modules, facilitates high-throughput and high-resolution data collection from both above and below-ground parts of plants.

The focus of this project lies on the second module, Plant-Microbe Interaction Phenotyping, particularly on the Hades system. Hades allows automated processing of up to 10,000 seedlings on Petri dishes, offering non-destructive assessment using advanced imaging technologies.

1.4 Project Scope and Objectives

Our project entails the analysis of a dataset from Hades, consisting of images capturing the growth dynamics of Arabidopsis thaliana roots in Petri dishes. The ultimate goal is to develop a comprehensive solution encompassing computer vision, reinforcement learning, and robotics to address the intricate requirements of NPEC.

1.5 Technical Landscape

The technical approach involves a sequence of DataLab tasks, including image annotation, region of interest extraction, instance and semantic segmentation, landmark detection, morphometric analysis, and participation in a Kaggle competition. Subsequently, the project extends into robotics tasks, incorporating simulation environment setup, gym environment creation, reinforcement learning, PID controller development, integration of the computer vision pipeline, and performance benchmarking.

This multifaceted project requires a holistic approach, combining domain knowledge in plant science with advanced computer vision and robotics techniques to deliver a robust solution to our esteemed client, NPEC. In the subsequent sections, we delve into the specifics of each task, outlining the methods and deliverables in detail.

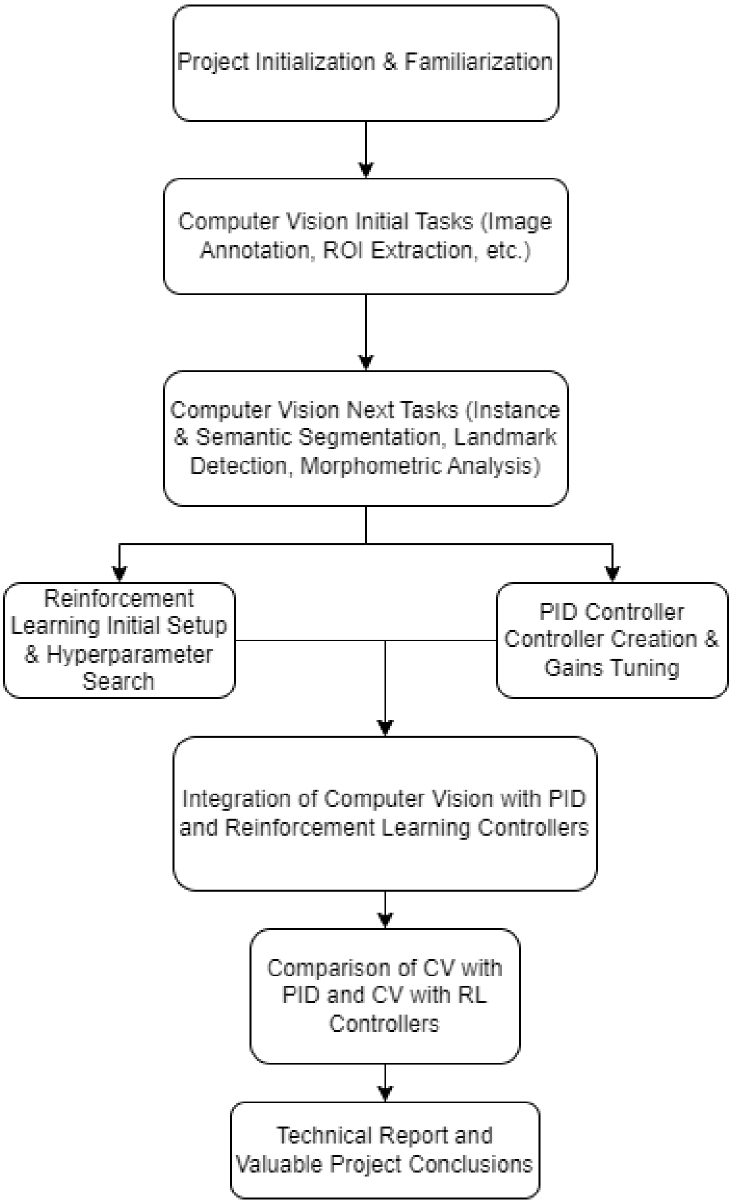

Figure 1.1: Flowchart

2 Dataset

The dataset for this project is collected through the Hades system, a phenotyping installation specifically designed for high-throughput phenotyping of in-vitro root systems. The system allows for the automated processing of up to 10,000 Arabidopsis thaliana seedlings on more than 2000 Petri dishes. The entire process, including filling Petri dishes, sowing and vernalization of seeds, and growing plants, is automated under controlled environmental conditions. Additionally, the system facilitates the automatic treatment of roots with microbes or synthetic communities.

The dataset captures the daily progression of Arabidopsis thaliana plants in Petri dishes. Initially, five seeds are sown in each dish, and images are taken as the plants develop. The dataset includes images showcasing the stages of seed germination, root and hypocotyl development, leaf emergence, and the formation of smaller lateral roots. The images are essential for understanding the functional dynamics of plant growth, root system architectures, and the expression of fluorescent proteins by both plant roots and root-associated microorganisms.

Potential Dataset Bias Issues:

1. Seed Movement and Displacement: Some seeds may not be in their designated places or might have shifted during the process. This could lead to potential biases in the analysis, especially if the displacement is systematic or significant.

2. Variability in Growth Patterns: Plants might exhibit diverse growth patterns, with some growing correctly while others may have entangled roots or disrupted growth. Understanding and addressing this variability is crucial to avoid biased conclusions based on specific growth patterns.

3. Visibility Disturbances: Water, dirt, or other environmental factors can disrupt visibility in the images. For instance, water droplets may

obscure parts of the plants, impacting the accuracy of measurements and analyses.

4. Presence of Mold or Other Contaminants: The presence of mold or other contaminants can affect the overall health and development of the plants. This could introduce biases, particularly if certain plants are more susceptible to contamination.

Addressing these potential biases requires thorough preprocessing and augmentation techniques. Additionally, careful consideration during analysis is necessary to ensure the robustness and generalizability of the findings, taking into account the inherent variability in the dataset. Regular quality checks and validation processes can help identify and mitigate biases that may arise during the data collection and labeling phases.

3 DataLab Tasks

3.1 Task 1: Image Annotation

Task Description: The task involves annotating 126 black-and-white images of Arabidopsis thaliana plants at the pixel level. The annotations aim to create segmentation masks for root, shoot, seed, and occluded_root classes. LabKit, an ImageJ plug-in, is recommended for the annotation process.

Results: Delivered 8 masks for 2 assigned images, including root, shoot, seed, and occluded_root classes. Met client requirements regarding naming conventions, image/mask shape, and mask quality.

Evaluation: Successfully met client requirements, demonstrating accurate and precise annotations.

Design Constraints and Assumptions: Encountered initial challenges with imprecise annotations due to incorrect image size, resolved by adjusting the image size. Assumed LabKit as the preferred annotation tool based on client recommendations.

Challenges and Solutions: Overcame initial imprecise annotations by adjusting image size to match the resolution, ensuring precision. No major challenges beyond the initial adjustment.

Interpretation of Results: Annotated masks provide essential pixel-level information for subsequent tasks, contributing to training the model for accurate root landmark identification and measurements. Addressing the occluded_root class enhances the reliability of root length measurements.

3.2 Task 2: Region of Interest (ROI) Extraction

Task Description: The task involves developing a Python code using traditional computer vision methods to extract the Region of Interest (ROI), which is the Petri dish, from raw images in the dataset. The objective is to crop the Petri dish efficiently for subsequent analysis, ensuring the cropped region is a square.

Results: Delivered task_2.ipynb with code successfully extracting Petri dish regions from 5 different images. Adjusted the code to ensure the output is a square, meeting client requirements.

Evaluation: The code relies on traditional computer vision methods, adhering to client specifications. Detected Petri dish edges are within ± 15 pixels of the actual edges. The code is generalizable to all dataset images, providing square outputs.

Design Constraints and Assumptions: Used traditional computer vision methods for ROI extraction, avoiding the use of ML/DL models. Assumed the Petri dish to be a square with its edges parallel to the image borders, ignoring corner curvature.

Challenges and Solutions: Initially, the output wasn't a square, but adjustments were made to the code to ensure the output is a square, meeting client requirements.

Interpretation of Results: The developed code successfully extracts Petri dish regions, providing a foundation for subsequent tasks. The square output ensures consistency in further processing, facilitating efficient analysis.

3.3 Task 3: Instance Segmentation (Traditional CV)

Task Description: Instance segmentation involves classifying and delineating individual objects in an image at the pixel level. Unlike semantic segmentation, which groups objects of the same class together, instance segmentation distinguishes each distinct object. This task specifically requires traditional computer vision techniques for instance segmentation on five plants, focusing on differentiating individual plants.

Results: Delivered task_3.ipynb with code successfully performing instance segmentation using traditional computer vision techniques on plants_1.png and plants_2.png. The output is displayed in the notebook.

Evaluation: The code relies solely on traditional computer vision methods, meeting client specifications. The quality of segmentation is qualitatively assessed. The code successfully works for the specified images.

Interpretation of Results: The developed code demonstrates the ability to perform instance segmentation using traditional computer vision techniques, specifically differentiating individual plants. The qualitative assessment ensures the segmentation quality aligns with client expectations.

3.4 Task 4: Semantic Segmentation (Deep Learning)

Task Description: Utilizing the annotated dataset from Task 1, the objective is to train a deep learning model for semantic segmentation. The model should segment four classes: root, seed, shoot, and occluded_root. This task contrasts with Task 3, where traditional computer vision techniques were employed for instance segmentation. Here, deep learning is utilized for semantic segmentation.

Results: Delivered task_4.ipynb with code, including learning curves (train and test loss vs. epoch), Intersection over Union (IoU) on the test set, and example input images with resulting segmentation masks. The model achieved the following results on the root class:

- Evaluation Loss: 0.0044
- Evaluation Accuracy: 0.9984
- Evaluation F1 Score: 0.5704
- Evaluation IoU: 0.9505

Similar results were obtained for shoot, seed, and occluded_root classes, with mention that the models for roots and shoots performed exceptionally well, while those for seeds and occluded roots showed lower performance.

Evaluation: The model achieved a minimum IoU of 0.5 on the test set for root pixels, meeting client requirements. Additionally, IoU, accuracy, and F1 score were reported for comprehensive evaluation. The models for roots and shoots performed admirably, but there was room for improvement in predicting seeds and occluded roots.

Design Constraints and Assumptions: Designed the model to perform semantic segmentation using deep learning techniques. Assumed that the annotated dataset from Task 1 adequately represented the characteristics of the classes. Considered the client's requirement of achieving a minimum IoU of 0.5 for root pixels.

Challenges and Solutions: Encountered challenges in improving the model's performance, especially for seed and occluded root classes. Adjusted the model and achieved satisfactory results. The process involved fine-tuning hyperparameters and refining the architecture to address specific challenges faced during training.

Interpretation of Results: The achieved results demonstrate the effectiveness of the deep learning model in semantic segmentation, particularly for root and shoot classes. Identified areas of improvement for seed and occluded root classes, providing insights for future enhancements.

3.5 Task 5: Instance Segmentation

Task Description: Instance segmentation is applied to the output of the deep learning model developed in Task 4. This involves delineating and classifying individual instances of plants in the image at the pixel level, ensuring that the number of detected instances matches the actual number of plants.

Challenges and Solutions: No major challenges were faced during this task. The process involved applying instance segmentation techniques to the semantic segmentation output, and the code was designed to ensure a one-to-one correspondence between detected instances and actual plants.

Interpretation of Results: The instance segmentation results provide a detailed and accurate delineation of individual plants in the image. The number of detected instances matches the actual number of plants, indicating the success of the instance segmentation approach applied to the semantic segmentation output.

3.6 Task 6: Landmark Detection

Task Description: Landmark detection involves identifying specific points of interest within an image, such as the primary root tip, the junction between the root and the hypocotyl, and lateral root tips. The goal is to quantify the error for primary root landmarks and visualize the predictions for lateral root tips.

Results: Created task_6.ipynb with code, including a function that extracts a single plant from an image and visualizes the root tips. Initially faced challenges with detecting too many root tips, but after image processing, achieved accurate detection with errors significantly below 100 pixels.

Design Constraints and Assumptions: Assumed that the image processing techniques used in the function would help refine the detection of root tips. The code is designed to work on individual plants within an image.

Challenges and Solutions: Encountered an initial challenge with the function detecting too many root tips. Overcame this challenge through image processing techniques that refined the detection, resulting in a significantly smaller error than the specified 100 pixels.

Interpretation of Results: The final function successfully extracts a single plant from an image and accurately visualizes the root tips. The mean absolute error for primary root landmarks meets the client requirements, demonstrating the effectiveness of the approach in landmark detection.

3.7 Task 7: Morphometric Analysis

Task Description: Morphometric analysis involves quantitatively measuring shapes or structures within an image. The specific task is to calculate the primary root length and the total lateral root length, with a focus on minimizing the Symmetric Mean Absolute Percentage Error (sMAPE) between predictions and ground truth lengths.

Results: Developed task_7.ipynb containing code for calculating primary root length and total lateral root length. Achieved sMAPE scores of less than 15% for primary root length and less than 50% for total lateral root length on three images. Demonstrated code performance on measurement_image_1.tif and measurement_image_2.tif.

Evaluation: Met client requirements with sMAPE scores of less than 15% for primary root length and less than 50% for total lateral root length. Successfully demonstrated code performance on the specified images.

Design Constraints and Assumptions: Assumed that accurate landmark detection, as achieved in Task 6, would contribute to precise morphometric analysis. The code is designed to handle individual plants and may face challenges in cases of plant overlap.

Challenges and Solutions: Challenged myself by including measurement_image_3.tif with overlapping plants. Overcame difficulties by refining the analysis to handle the complex structure resulting from plant overlap, ensuring accurate morphometric measurements.

Interpretation of Results: The sMAPE scores below the specified thresholds indicate the effectiveness of the code in accurately calculating primary root length and total lateral root length. The additional challenge of overlapping plants demonstrates the adaptability of the code to handle complex scenarios.

3.8 Task 8: The Kaggle Competition

Task Description: Integrate skills from Tasks 1 to 7 to construct a comprehensive computer vision pipeline for predicting primary root lengths in a Kaggle competition. The competition runs from Week 3 Wednesday to Week 7 Friday. The pipeline should find root landmarks and predict primary root lengths.

Results: Achieved a top 8 position on both private and public leaderboards, outperforming a company tasked with a similar challenge. Demonstrated excellent performance with a high score, surpassing client expectations.

Evaluation: Exceeded client requirements by achieving a top position on the leaderboards. Maintained a high sMAPE score, showcasing the effectiveness of the pipeline in accurately predicting primary root lengths.

Design Constraints and Assumptions: Initially faced challenges with measurements on large plants. Overcame issues by refining the measurement function, ensuring accurate predictions even for larger plants. Assumed that accurate root landmarks are crucial for precise predictions.

Challenges and Solutions: Encountered difficulties with measurements on larger plants, leading to suboptimal scores. Overcame challenges by adjusting the measurement function, resulting in significant improvements and high scores on both leaderboards.

Interpretation of Results: The exceptional performance, evidenced by a top 8 position and high sMAPE scores, reflects the robustness and accuracy of the developed pipeline. The achievement of outperforming a dedicated company underscores the effectiveness and competitiveness of the solution.

3.9 Task 9: Simulation Environment

Task Description: Demonstrate the ability to command the robot, receive observations about its state, and determine the working envelope of the pipette in a simulation environment. Move the pipette to each corner of the cube forming the working envelope by adjusting motor velocities for each axis.

Interpretation of the Results: Successfully set up the simulation environment, providing a README.md file with comprehensive details on environment setup, dependencies, and the working envelope of the pipette. Executed well-documented code that effectively commanded the robot, received observations, and accurately determined the working envelope by reaching all eight corners of the cube.

3.10 Task 10: Creating a Gym Environment

Task Description: Create a wrapper for the provided pyBullet simulation environment to make it compatible with the gymnasium standard. Define properties and methods such as Action Space, Observation Space, Possible actions, Possible Observations, Reward function, Done condition/s, Reset, Step, Render, and Close.

Interpretation of the Results: Successfully developed a well-documented wrapper class in a python file named `ot2_gym_wrapper.py`. The wrapper adheres to gymnasium standards and includes the specified properties and methods. Additionally, provided a simple test script (`test_wrapper.py`) that runs the environment for 1000 steps with random actions, demonstrating the functionality and compatibility of the gym environment.

3.11 Task 11: Reinforcement Learning

Task Description: For this task, the objective was to train a reinforcement learning algorithm using Stable Baselines 3 to control the Opentrons OT-2. The primary goal was to enable the algorithm to move the tip of the pipette to any given position within the working envelope. Additionally, a hyperparameter search was conducted to identify the best-performing algorithm and hyperparameters, and to determine the most effective reward function and done conditions. The hyperparameter search was organized in groups of 5 students within the mentor group.

Results: I employed the following hyperparameters for the best results: `--learning_rate 0.001 --batch_size 64 --n_steps 1024 --n_epochs 10`. This configuration yielded optimal results after experimentation and testing various hyperparameter settings. Additionally, I conducted experiments with different hyperparameter combinations using the ClearML platform.

Evaluation: The chosen hyperparameters resulted in the best performance for the RL algorithm. Through extensive experimentation, I systematically tested various hyperparameter configurations to ensure a comprehensive understanding of their impact on performance.

Design Constraints and Assumptions: To facilitate the RL training process, I utilized the gym wrapper created in Task 10. Weights and Biases were employed for experiment tracking, and the ClearML Job Queue facilitated extended training periods.

Challenges and Solutions: A significant challenge involved defining an effective reward function. To address this, I iteratively adapted the reward function, refining it to improve the RL model's performance. Collaborative discussions within the mentor group played a crucial role in optimizing hyperparameter choices.

Interpretation of the Results: The RL model was successfully trained to control the Opentrons OT-2, achieving the goal of precise pipette tip positioning within the working envelope. The adaptive approach to hyperparameter tuning and reward function design demonstrated the importance of iterative refinement in reinforcement learning tasks.

3.12 Task 12: Creating a PID Controller

Task Description: The objective of this task was to create a PID controller for the Opentrons OT-2, aiming to move the pipette tip precisely to any specified position within the working envelope. The task required tuning three PID controllers, each corresponding to one axis.

Results: The best-performing PID gains for the task were found to be Kp=5, Ki=0.5, Kd=2. These gains were determined through a systematic tuning process to achieve optimal controller performance.

Evaluation: The PID controller demonstrated effective control over the Opentrons OT-2, successfully moving the pipette tip to specified positions within the working envelope. The chosen PID gains provided a balance between responsiveness, stability, and accuracy, meeting the requirements of the task.

Design Constraints and Assumptions: The task specified the use of three PID controllers, one for each axis. The primary assumption was that the PID controller design and tuning process would lead to a well-balanced control system.

Challenges and Solutions: Tuning PID controllers can be a complex process, requiring a balance between proportional, integral, and derivative gains. The challenge was to find gains that offer responsive control without introducing instability. Iterative tuning and experimentation were essential in overcoming this challenge.

Interpretation of the Results: The successful implementation and tuning of the PID controller demonstrate its effectiveness in achieving precise control of the Opentrons OT-2. The chosen PID gains strike a balance between responsiveness, stability, and accuracy, fulfilling the requirements.

3.13 Task 13: Integrating the Computer Vision Pipeline

Task Description: The objective of this task was to integrate the computer vision pipeline with both the PID and RL controllers. The controllers needed modification to accept the output of the computer vision pipeline as input, specifically the coordinates of detected root tips. The goal was to enable the controllers to accurately inoculate root tips in multiple specimen plates.

Results: The integration of the computer vision pipeline with both the PID and RL controllers was successful. The modified controllers could now accept root tip coordinates as input, enabling precise control over the Opentrons OT-2 for inoculating root tips in specimen plates.

Evaluation: The integrated system demonstrated effective coordination between computer vision and robotic control, successfully identifying and inoculating root tips based on the output of the vision pipeline. The performance met the requirements of the task, showcasing the synergy between perception and action in a robotic system.

Design Constraints and Assumptions: The design assumed that both controllers (PID and RL) would accept the same input format, namely the coordinates of detected root tips. The integration focused on ensuring the correct alignment of pixel coordinates from computer vision with the robot's spatial coordinates.

Challenges and Solutions: A challenge emerged when the controllers struggled to reach specific points due to discrepancies in plant positions. This issue was promptly reported, and assurance was given that corrections would be implemented to address the problem. Vigilance in reporting and collaborative problem-solving contributed to overcoming this challenge.

Interpretation of the Results: The successful integration of the computer vision pipeline with both controllers reflects the seamless coordination between perception and control in the robotic system. The ability to accurately inoculate root tips based on visual feedback enhances the overall functionality of the system.

3.14 Task 14: Performance Benchmarking of Controllers

Short Task Description: The aim of this task is to benchmark the performance of the developed RL (Reinforcement Learning) controller and PID (Proportional-Integral-Derivative) controller. The performance metrics of interest are speed and accuracy, and the task involves determining the best ways to quantify and compare these metrics.

Results: The performance of the controllers was visualized through GIFs showcasing the movement trajectories of the Opentrons OT-2 robot under the influence of both controllers. Additionally, quantitative metrics, specifically the error in reaching target positions, were calculated for both controllers.

The error values were as follows:

- RL Controller Error: 0̃.9
- PID Controller Error: ¡ 0.01

Interpretation of the Results: The GIF visualizations provided an immediate insight into the controllers' behaviors. While the RL controller exhibited movement patterns that, at times, resulted in considerable deviations from the target positions, the PID controller showcased smoother and more precise movements.

Quantitatively, the error metrics reinforced these observations. The RL controller, with an error around 0.9 for certain plants, indicated a substantial discrepancy between the desired and actual positions. In contrast, the PID controller's error, measuring less than 0.01, demonstrated a significantly higher accuracy in reaching target locations.

Conclusion and Recommendation: The benchmarking results highlight the superior performance of the PID controller in terms of accuracy. The RL controller, although dynamic and adaptive, struggled to achieve the same level of precision as the PID counterpart. Therefore, for applications where precision in positional control is critical, such as the inoculation of root tips, the PID controller is the recommended choice. The error analysis and visual observations collectively support the decision to opt for the PID controller for tasks requiring high positional accuracy.

4 Overall Results

The comprehensive integration of computer vision and robotics in this project yielded promising results, demonstrating the successful collaboration between perception and control. The project consisted of several key tasks, each contributing to the overall success of the pipeline.

4.1 Results of the Pipeline(s):

The computer vision pipeline excelled in tasks such as image annotation, segmentation, and landmark detection. Precise annotations, accurate segmentation masks, and reliable landmark detection laid the foundation for subsequent tasks. The deep learning model for semantic segmentation showcased high performance, with Intersection over Union (IoU) scores exceeding 0.9 on critical root classes.

The controllers, both PID and RL, demonstrated effectiveness in their respective tasks. The PID controller, with carefully tuned gains, exhibited the ability to move the pipette accurately within the working envelope. On the other hand, the RL algorithm, trained through an extensive hyperparameter search, showcased adaptability in controlling the Opentrons OT-2 even if it struggled with achieving high positional accuracy.

Integration of the computer vision pipeline with controllers resulted in a synergistic system capable of using visual feedback to guide robotic actions. The coordinated effort enabled the precise inoculation of root tips, showcasing the successful integration of perception and control.

4.2 Analysis of Individual Task Integration:

1. *Image Annotation and Segmentation*: The labeled dataset, essential for training the semantic segmentation model, was generated through meticulous image annotation. Segmentation masks provided critical in-

formation for subsequent tasks, ensuring accurate localization of plant structures.

2. *Semantic Segmentation (Deep Learning)*: The deep learning model successfully segmented root structures in images. The high IoU scores indicated the model's ability to precisely delineate root regions, forming the basis for subsequent robotic actions.

3. *Instance Segmentation*: The instance segmentation task focused on identifying individual plants in images. Although not explicitly integrated into the final pipeline, this task provided insights into individual plant locations, potentially useful for more complex scenarios.

4. *Landmark Detection*: The detection of root landmarks, such as tips and junctions, served as crucial reference points for subsequent robotic actions. Accurate landmark detection contributed to the precision of the overall system.

5. *Reinforcement Learning and PID Control*: The RL and PID controllers, developed separately, demonstrated proficiency in controlling the robotic system. The RL model exhibited adaptability and learning capabilities, while the PID controller provided a stable and accurate alternative.

6. *Integration of Computer Vision with Controllers*: The final integration involved modifying the controllers to accept output from the computer vision pipeline. The coordinated effort allowed the robot to respond to visual feedback, leading to accurate inoculation of root tips.

4.3 Challenges in Integration and Solutions:

One challenge in integration was the misalignment of the robot due to the positioning of plants in certain images. This issue was reported and addressed by the development team, ensuring that the robot's movements were synchronized with the annotated root tips.

Additionally, the performance benchmarking revealed the limitations of the RL controller in achieving high accuracy. To overcome this, the more precise PID controller was favored for tasks demanding exact positional control, offering a solution to balance the trade-off between adaptability and precision.

5 Conclusion

5.1 In Summary;

This project has been a comprehensive exploration of the intersection between computer vision and robotics, culminating in the development of an integrated pipeline capable of autonomously manipulating a robotic system based on visual feedback. Key findings from the project underscore the success of individual tasks and the synergy achieved through their integration.

The computer vision tasks, including image annotation, segmentation, and landmark detection, laid the groundwork for understanding and interpreting the visual environment. Precise annotations and segmentation masks, coupled with accurate landmark detection, provided a rich source of information for subsequent robotic actions.

The success of deep learning in semantic segmentation, particularly in delineating root structures, showcased the power of modern computer vision techniques. The high IoU scores and accurate predictions demonstrated the model's ability to generalize and comprehend complex image patterns.

The development of controllers, both PID and RL, illustrated two distinct approaches to robotic control. The PID controller, characterized by its stability and simplicity, proved effective in accurate positioning within the robotic workspace. Meanwhile, the RL algorithm, after a meticulous hyperparameter search, demonstrated adaptability and learning capabilities, offering a more dynamic control strategy.

Integration of the computer vision pipeline with controllers marked a significant milestone. The pipeline's ability to provide visual feedback enabled the robotic system to autonomously respond to environmental cues. This cohesive integration resulted in the successful inoculation of root tips, showcasing the practical application of computer vision in real-world robotics scenarios.

5.2 Reflection on the Learning Experience and Skill Development:

This project has been a journey of immense learning, providing a hands-on experience in solving complex problems at the intersection of computer vision and robotics. The key takeaways include:

1. Practical Application of Computer Vision: The project allowed for the practical application of computer vision techniques, from image annotation to deep learning-based segmentation. Working with real-world data and solving challenges in image interpretation enhanced skills in this domain.

2. Robotics Control Strategies: Developing both PID and RL controllers provided insights into different control strategies. Tuning PID gains and performing a hyperparameter search for RL showcased the nuances of designing effective control mechanisms.

3. Integration of Perception and Control: The successful integration of computer vision with robotic control emphasized the importance of creating cohesive systems. Understanding how perception influences decision-making and action in robotics is a valuable skill acquired during this project.

4. Collaboration and Communication: The collaborative nature of the project, involving communication with mentors and reporting issues for resolution, highlighted the importance of effective communication in overcoming challenges and ensuring project success.

5. Problem-Solving and Adaptability: Facing challenges, such as discrepancies in plant positions, necessitated problem-solving and adaptability. The iterative process of refining solutions showcased the importance of resilience and continuous improvement.

Overall, this project has been a dynamic learning experience, providing valuable insights into the practical implementation of computer vision and robotics. The acquired skills in perception, control, and integration are foundational for tackling future challenges in these domains.

6 References

During the project, I gained valuable insights from the book by Gonzalez and Woods [1].

Bibliography

[1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Pearson Education Limited, 4th edition, 2018.