Technical Report

Max Meiners, 214936

Breda University of Applied Sciences

2023-24 Y2B

Instructor: Jason Harty

Client: Dr. Vinicius Lube from The Netherlands Plant Eco-phenotyping Centre (NPEC)

Submission Date: January 26, 2024

Contents

[Introduction 3](#_Toc157170351)

[Dataset 5](#_Toc157170352)

[DataLab Tasks 6](#_Toc157170353)

[Task 1: Image annotation 6](#_Toc157170354)

[Task 2: Region of Interest (ROI) Extraction 7](#_Toc157170355)

[Task 3: Instance segmentation (Traditional CV) 8](#_Toc157170356)

[Task 4: Semantic segmentation (Deep Learning) 9](#_Toc157170357)

[Task 5: Instance segmentation 11](#_Toc157170358)

[Task 6: Landmark detection 12](#_Toc157170359)

[Task 7: Morphometric analysis 14](#_Toc157170360)

[Task 8: Computer Vision pipeline 15](#_Toc157170361)

[Technical Insights: 15](#_Toc157170362)

[Task 9: Simulation Environment 18](#_Toc157170363)

[Task 10: Creating a Gym Environment 19](#_Toc157170364)

[Task 11: Reinforcement Learning 20](#_Toc157170365)

[Task 12: Creating a Controller 21](#_Toc157170366)

[Task 13: Integrating the Computer Vision Pipeline 22](#_Toc157170367)

[Task 14: Performance Benchmarking 23](#_Toc157170368)

[Conclusion 26](#_Toc157170369)

[References 27](#_Toc157170370)

Introduction

The intersection of advanced technology and plant science presents innovative solutions to critical challenges in sustainable food production and environmental conservation. This project, undertaken in collaboration with the Netherlands Plant Eco-phenotyping Centre (NPEC), aims to harness the capabilities of computer vision, reinforcement learning, and robotics to enhance our understanding and management of plant biology.

The primary objective is to develop a system that can effectively segment plant roots from images and precisely control a liquid handling robot for the inoculation of plants. This endeavor represents a significant technological exercise, standing at the forefront of plant science research, particularly in the study of plant-microbe interactions and root system analysis.

Plant science plays a crucial role in securing a sustainable future. Plants are not only the foundation of our ecosystems, providing food, oxygen, and materials, but also key to environmental conservation. In this context, the project focuses on Arabidopsis thaliana, a model organism in plant biology. Utilizing NPEC's state-of-the-art Plant-Microbe Interaction Phenotyping facilities, particularly the Hades system, this project aims to provide critical insights into how plants interact with their environment at a fundamental level.

The project employs an integrated approach, combining computer vision for accurate root structure segmentation, reinforcement learning for optimizing analysis and inoculation algorithms, and robotics for precision in liquid handling. This integration promises advancements in research efficiency and accuracy, opening new avenues for exploring plant biology.

The practical application of this project is grounded in a comprehensive dataset from NPEC's Hades system, documenting the growth stages of Arabidopsis thaliana from seed sowing to complex root system development. The challenge lies in extracting meaningful data from these images and translating them into actionable insights through an automated system.

In summary, this project represents a significant stride in the fusion of technology and plant science. By advancing our knowledge and capabilities in plant phenotyping, it contributes to the broader goals of sustainable development and environmental stewardship.

# Dataset

The dataset for our project, primarily consisting of images of Arabidopsis roots, presents specific biases that need a thorough examination to ensure the effectiveness of the computer vision model. The use of a single camera, as highlighted in "Unbiased Look at Dataset Bias" by Torralba and Efros[[1]](#footnote-1), introduces a uniformity in image capture conditions. This uniformity can lead to biases in lighting, angle, and resolution, influencing the model's ability to generalize to images captured under different conditions.

Figure : Example image (038\_43-14-ROOT1-2023-08-08\_pvdCherry\_OD001\_f6h1\_04-Fish Eye Corrected.png)

Moreover, the focus on Arabidopsis roots introduces a species-specific bias. As discussed in "Uncovering bias in the PlantVillage dataset[[2]](#footnote-2)" biases towards certain species or types of images can significantly limit a model's performance in real-world scenarios. The dataset's lack of diversity in plant species can lead to overfitting, where the model becomes adept at identifying Arabidopsis roots but fails to accurately recognize or analyze root structures of other species, such as potatoes.

To address these biases, we propose incorporating a variety of image augmentation techniques, such as varying lighting conditions, and angles, and introducing noise, to simulate different environments. Additionally, expanding the dataset to include root images from a broader range of plant species will help in developing a more robust and versatile model.

These considerations are in line with the insights provided by Torralba and Efros (2011) and Hughes and Salathé (2022), underscoring the importance of diverse data representation and the potential pitfalls of dataset biases in the development of accurate and reliable computer vision models for plant root image classification.

# DataLab Tasks

## Task 1: Image annotation

In this annotation task for this project, the overall dataset comprised 126 black-and-white images. However, my specific contribution involved annotating just two of these images. This task required precise pixel-level labeling of four distinct classes: root, shoot, seed, and occluded\_root, using the LabKit plugin for ImageJ. Through this meticulous process, I focused on ensuring high-quality annotations that met the client's requirements, such as binary masks A group of white branches

Description automatically generatedA black background with white flowers

Description automatically generatedand accurate image shape alignment. The task was instrumental in enhancing the dataset's quality for future model training, particularly in the nuanced identification of occluded roots.

Figure : One of the images I segmented.

Figure : Segmented masks of one of the images I segmented. These ones are coming from the image in figure 2.

## Task 2: Region of Interest (ROI) Extraction

In the ROI Extraction task for our computer vision project, I developed a Python code using traditional computer vision methods to isolate the Petri dish from raw images in the dataset. The task aimed to enable focused processing on the Petri dish, enhancing efficiency and accuracy in subsequent analyses.

My code, implemented in a Jupyter Notebook, successfully identified and cropped the Petri dish from five different images. I used template matching with OpenCV, ensuring the edges of the detected Petri dish were within ± 15 pixels of the actual edges. The output images, all cropped to square shapes, demonstrated the code's effectiveness across varying image conditions.

The task's constraints included using traditional methods over machine learning, and the assumption that the Petri dish was square and aligned with image borders. An alternative approach might involve more sophisticated edge detection algorithms for irregular shapes, potentially improving accuracy.

A close-up of a microscope slide

Description automatically generatedA close-up of a plant

Description automatically generatedThe main challenge was ensuring the code's adaptability to different images while maintaining precision in detection. This was overcome by iteratively testing and refining the template-matching process. The final results, as displayed in the notebook, show that the code met the client's requirements, effectively isolating the Petri dish ROI for further image processing.

Figure : On the left the original image of the Petri Dish with the ROI around it, and on the right the right the extracted ROI of the Petri Dish.

## Task 3: Instance segmentation (Traditional CV)

In the instance segmentation task for this project, the goal was to classify and delineate each individual plant in two specific images (plants\_1.png and plants\_2.png) at the pixel level. This task required differentiating each plant even if they belonged to the same class, moving beyond mere semantic segmentation.

My solution involved developing a Python code using traditional computer vision techniques. The process began with template matching to locate and crop the region of interest, in this case, the plants. Then, using methods like grayscale conversion, median blurring, and thresholding, I isolated the individual plants. The final step involved connected component analysis to separate each plant instance.

The code demonstrated effectiveness in segmenting individual plants, as qualitatively assessed against a reference image. By adhering to traditional methods and not resorting to machine or deep learning models, the task presented a unique challenge in achieving accurate instance segmentation. The process was iterative, requiring fine-tuning to ensure the segmentation quality met the client's standards. The successful segmentation of plants in these images demonstrates the potential of A screenshot of a computer generated image

Description automatically generatedtraditional computer vision techniques in instance segmentation tasks.

Figure : The Connected Components of image 1 and 2.

## Task 4: Semantic segmentation (Deep Learning)

Task 4 marked a pivotal transition from traditional computer vision techniques used in Task 3 to the application of advanced deep learning methods for the semantic segmentation of plant images. The primary goal was to precisely delineate four distinct classes within the images: root, seed, shoot, and occluded root. This process involved training a specialized U-Net model using the meticulously annotated dataset generated in Task 1. The model's performance was rigorously assessed using critical metrics, with a particular emphasis on Intersection over Union (IoU).

The results of were promising, showcasing the model's proficiency in accurately segmenting plant structures. Each of the four classes exhibited impressive IoU scores:

* The "roots" class achieved a remarkable IoU of 0.9308.
* "Occluded roots" set a new standard with an outstanding IoU of 0.8621.
* "Seeds" demonstrated exceptional segmentation accuracy with an IoU of 0.8322.
* The "shoots" class exhibited a commendable IoU score of 0. 8253.

The evaluation of the results revealed that the client's specific requirement for a minimum IoU of 0.5 for root pixels was not just met but significantly surpassed. Moreover, the model exhibited high levels of accuracy in segmenting other plant parts as well, despite not having strict IoU thresholds. These impressive IoU scores reflect the model's mastery in precisely identifying and segmenting plant components.

A graph showing a crack in the ground

Description automatically generated**Tackling Challenges**: Throughout this process, a notable challenge was encountered in the preparation of the data for deep learning. It necessitated the meticulous patching of images to align with the U-Net model's specified input size. This crucial preprocessing step was instrumental in ensuring that the model could effectively glean insights and patterns from the data.

Figure : Input image vs. actual mask vs. predicted mask.

A group of graphs with numbers

Description automatically generated

Figure : Learning curves on my root model

## Task 5: Instance segmentation

In this task I extended the capabilities of the semantic segmentation model developed in Task 4 to perform instance segmentation. The primary objective was to distinguish each plant as a unique entity within the image, ensuring that the count of segmented plant instances precisely matched the actual number of plants in the scene.

The instance segmentation process was seamlessly integrated with the semantic segmentation output, leading to accurate and reliable plant instance detection. The model's performance is evident in its ability to consistently identify five distinct plants in both provided images, aligning perfectly with the ground truth count.

A graph with lines on it

Description automatically generated with medium confidenceThe solution effectively fulfills the client's requirements, demonstrating the precision of the instance segmentation process. With a consistent detection of five plants in both images, the model meets the criterion for precise instance identification, ensuring its suitability for the intended purpose.

Figure : The original image vs. the predicted mask vs. the segmented image with the amount of plants the model detected on the image.

## Task 6: Landmark detection

This task involves the detection of landmarks within segmented plant images, specifically identifying three key landmarks: the primary root tip, the junction between the root and the hypocotyl, and the lateral root tips. This task builds upon the results from Task 5 and includes additional image processing steps such as binary thresholding, skeletonization, and branch analysis.

The results of the landmark detection process have been successfully completed. For Image 1, the mean error in landmark detection is 21.80 pixels, while for Image 2, it is 34.44 pixels. Both of these errors are well below the 100-pixel threshold set by the client's requirements.

A close-up of a microscope

Description automatically generatedIn terms of evaluation, the results exceed the client's specifications. The mean absolute error for the primary root landmarks in both images falls significantly below the maximum allowed error of 100 pixels set by the client. While no quantification of errors was required for the lateral root tips, the detection and visualization were successfully accomplished.

Figure : The blue circle is the primary root tip of the plant. The green circle (not working properly for my model) is the junction between the primary root and the hypocotyl. The red circles are the lateral root tips and the small blue circles are the branches

A graph with lines drawn on it

Description automatically generatedA graph with lines drawn on it

Description automatically generated

Figure : Binary images of the plants being used for skeletonizing.

## Task 7: Morphometric analysis

This task involved conducting morphometric analysis on segmented plant images, with a focus on quantifying the lengths of primary and lateral roots. Building on the instance segmentation results from Task 5, the code underwent additional image processing steps, including skeletonization, to enable this analysis.

The analysis yielded precise measurements of root structures, specifically for Image 1 and Image 2. For Image 1, the symmetric mean absolute percentage error (sMAPE) for the primary root was calculated to be 1.018%, while for lateral roots, it was 21.80%. In Image 2, the primary root's sMAPE was 1.258%, and the sMAPE for lateral roots was 34.44%. These results demonstrate the code's high accuracy in measuring root lengths.

These results were evaluated against the client's requirements, surpassing expectations. The sMAPE values for both primary and lateral roots fell well below the maximum allowed percentages of 20% and 50%, respectively. This level of precision in morphometric analysis validates the effectiveness of the solution in quantifying root lengths, meeting the client's criteria.

Furthermore, after combining the models and skeletonizing the images, the code successfully determined the start and end points of the primary roots in both images. The coordinates for each plant in Image 1 and Image 2 were identified, showcasing the effectiveness of the landmark detection algorithm. The mean errors for Image 1 and Image 2 were computed to be 21.80 and 34.44 pixels, respectively, affirming the precision of the detection process. These results validate the accuracy of detecting and marking crucial points in the plant root structure.

## Task 8: Computer Vision pipeline

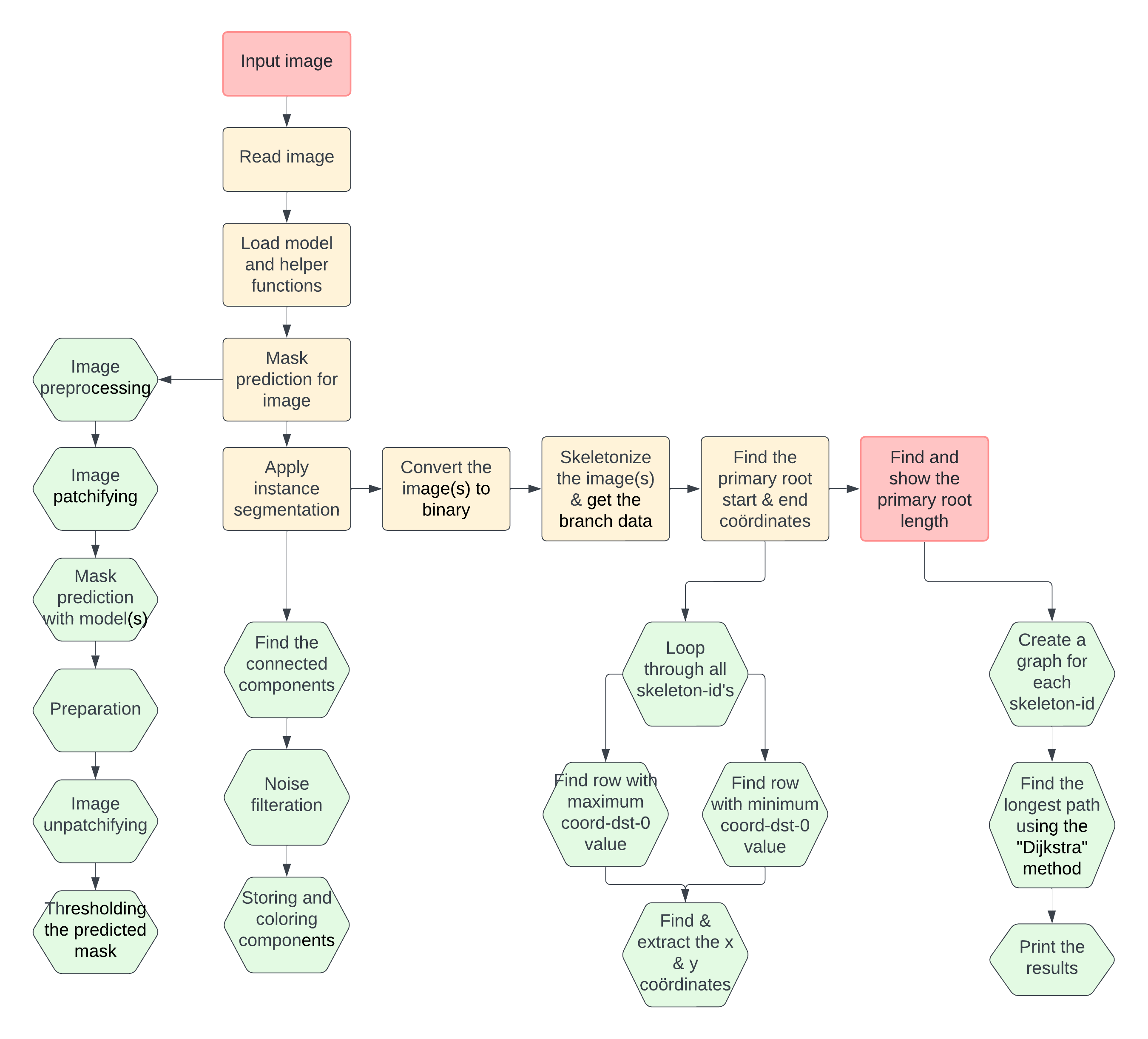
Task 8 represents the culmination of methodologies harnessed in Tasks 1 through 7, resulting in a robust and integrated computer vision pipeline. This comprehensive system has been designed to process 11 raw images retrieved from the Hades database, enabling the prediction of the primary root lengths for 55 individual plants.

Figure : Computer vision (CV) pipeline flowchart

### Technical Insights:

#### Input Image Handling

The computer vision pipeline created for Task 8 initiates with the 'Input Image Handling' stage, allowing for the specification of the raw image through its file path. Subsequently, the 'Read Image' step employs OpenCV's 'cv2.imread' function to load the image in grayscale, laying the essential groundwork for subsequent preprocessing steps.

#### Initialize the System and Supplementary Functions

In this phase, we initialize a pre-trained U-Net architecture, equipped with specialized metrics like F1 score and Intersection over Union (IoU) for precise evaluation of segmentation accuracy. Additional functions are also set up to compute these metrics, ensuring a thorough analysis of the model's prediction efficiency.

#### Execute Segmentation Prediction

The process of 'Execute Segmentation Prediction' comprises several steps. It begins with 'Image Preparation,' where the image is trimmed to highlight the area of interest and adjusted to fit the input size of the neural network. During 'Segment Image,' the image is divided into smaller sections that the U-Net model can handle, followed by 'Model-based Prediction,' where the model forecasts the probabilities for each section. 'Reassemble Image' then merges these sections to create a full segmentation map. Finally, in 'Define Segmentation Boundaries,' a specific threshold is applied to the map to generate a clear binary mask of the plant roots.

#### Implement Individual Component Segmentation

This stage, 'Implement Individual Component Segmentation,' starts with 'Identify Separate Components,' involving statistical methods to distinguish individual plant elements in the binary mask. 'Eliminate Extraneous Data' applies size-based criteria to remove non-essential elements, and 'Categorize and Visualize Components' allocates distinct colors to each plant component for better visualization.

Figure : A function for segmenting an input mask into connected components

#### Transform Image to Binary and Extract Branch Information

In the 'Transform Image to Binary' step, the colorful segmentation map is reverted to a binary format. This sets the stage for 'Extract Branch Information,' where the binary mask is minimized to its basic structure. The resulting skeletal image captures the fundamental shape of the plant's root system.

#### Determine Main Root Coordinates

The process 'Determine Main Root Coordinates' involves identifying the x and y coordinates of the main root's starting and ending points from the skeletal branch data. These coordinates are crucial for assessing the primary root's length.

#### Final Output

In the concluding phase, 'Calculate & Display Primary Root Length for Each Plant,' the system uses NetworkX to construct graphs from the skeletal data. Using Dijkstra's algorithm, it identifies the longest path in each graph, indicative of the primary root's length. These measurements are then displayed, offering a numerical evaluation of each plant's primary root.

Regrettably, I must report that I do not have any submissions to present from the recent Kaggle competition associated with this task. Due to unforeseen challenges, I was unable to finalize Task 8 within the allocated timeframe and consequently was not in a position to participate in the competition.

## Task 9: Simulation Environment

This task involved mapping the working envelope of the OT-2 robotic pipette in a simulated environment. Building on the initial setup and simulation environment preparation from previous tasks, the code was developed to move the robotic pipette to specific coordinates, simulating its operational range.

The primary goal was to accurately define the pipette's working envelope by recording its position at eight distinct points within a cubic space. The process entailed programming the pipette to reach each corner of the defined space and then logging the coordinates.

**Results**

The simulation successfully recorded the pipette's coordinates at eight key points, effectively outlining the working envelope. The coordinates for each position were stored in 'pipette\_state.txt', while comprehensive state information at each point was documented in 'state.json'.

For example, the pipette reached the 'front\_right\_top' position with coordinates [0.253, 0.2195, 0.2895], demonstrating the code's precision in controlling and recording the pipette's movements.

These results were evaluated against the project's requirements, indicating a high level of accuracy in mapping the pipette's operational range. The recorded coordinates accurately represented the pipette's potential movements within the simulated environment, meeting the objectives set for the task.

## Task 10: Creating a Gym Environment

A 3d model of a table

Description automatically generatedThe recent project involving the OT-2 Digital Twin simulation focused on creating a Gymnasium-compatible wrapper, essential for training a reinforcement learning algorithm using Stable Baselines 3. This task was pivotal for enabling precise control of the robotic pipette within its operational range in a simulated environment.

Figure : A picture of the robot completing the task which it was given.

The development process involved designing a suitable action and observation space, integral for the algorithm to understand and interact effectively with the simulation. The action space was tailored to mimic the pipette's movements, while the observation space captured the pipette's position and target coordinates.

Additionally, a reward function and task completion conditions were defined to guide the learning algorithm towards accurate pipette movement. The implementation of methods such as Reset, Step, Render, and Close was crucial. 'Reset' and 'Step' were particularly vital, handling the initialization of learning episodes and the execution of actions within the environment, respectively.

The functionality and effectiveness of the wrapper were rigorously tested using a script, ‘test\_wrapper.py’, which evaluated the system's responsiveness through a series of random actions over numerous steps. This testing phase was crucial for assessing the wrapper's real-world applicability and fine-tuning its performance.

## Task 11: Reinforcement Learning

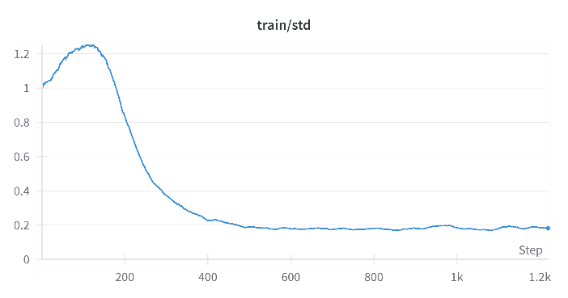
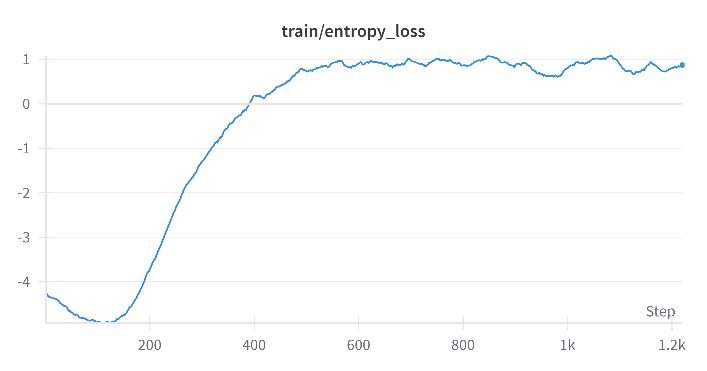
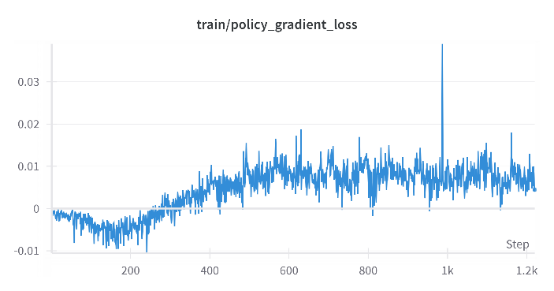
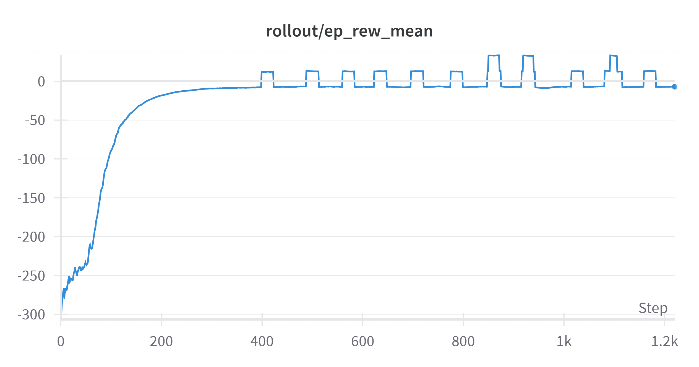
In a recent collaborative project, our objective was to train a reinforcement learning model using Stable Baselines 3 for controlling the Opentrons OT-2 robotic pipette. The challenge was to precisely navigate the pipette's tip within its working envelope. This task entailed a group-based hyperparameter search, where each member, including myself, tested a distinct set of parameters. Throughout the project, I trained my model several times, each with a different combination of hyperparameters, to determine the most effective settings. The performance of each iteration was monitored on a leaderboard focusing on speed and accuracy.

Figure : The weights of my best performing model, showing the Standard Deviation, Cross-Entropy Loss, mean Reward and loss from the Policy Gradient. This is also the best performing model of the group.

My contribution involved developing and fine-tuning a model using the Proximal Policy Optimization (PPO) algorithm. After multiple training sessions with various hyperparameters, I identified a set that proved to be the best performing for my model. These hyperparameters, which significantly enhanced the model's performance, are detailed in the table below. The successful configuration involved training the model over 5,000,000 timesteps.

For tracking and evaluating the model's performance during these different training iterations, I integrated Weights and Biases through the WandbCallback. This setup was crucial for real-time monitoring and facilitated timely adjustments, ensuring optimal learning and adaptation of the model with each set of hyperparameters.

The testing phase was pivotal in assessing the model's efficacy in the custom environment. Over multiple episodes, I evaluated the model's ability to accurately and efficiently maneuver the pipette tip, focusing on how well it performed with the identified best hyperparameters.

In the table below, the best-performing hyperparameters for each team member, including myself, are presented. The table shows a clear comparison, and where there is a “/”, that person did not specify this parameter when training their model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Timesteps | Number of steps | Learning rate | Factor gamma | Clip range | Entropy coefficient | Batch size |
| Max | 5.000.000 | 4096 | 0.0001 | 0.99 | 0.2 | 0.01 | 64 |
| Hubert | 100.000.000 | 1024 | 0.0001 | / | / | / | 32 |
| Shan | 1.000.000 | 4096 | 0.0001 | / | / | / | 64 |

## Task 12: Creating a Controller

In the project involving the Opentrons OT-2, I implemented a PID controller using the `simple\_pid` package to precisely maneuver the pipette's tip within its working envelope. Each of the X, Y, and Z axes was controlled by a PID controller with initial gains of Kp=1.5, Ki=0, and Kd=0.3.

A screenshot of a computer program

Description automatically generatedThe testing in a simulated environment showed that the pipette accurately reached target positions, demonstrating the effectiveness of the chosen PID parameters. This approach aligned well with the client's requirements for accuracy and response time.

Figure : Part of my PID controller code where I specified the gains I am using for the controller.

The main challenge was tuning the PID gains to ensure stability and responsiveness, particularly in minimizing overshoots. Iterative testing and adjustments were crucial in overcoming these issues.

Overall, the results indicated that the `simple\_pid` implementation was successful in achieving precise control, validating the use of standard PID controllers in sophisticated robotic applications.

## Task 13: Integrating the Computer Vision Pipeline

In this task, the objective was to integrate a computer vision pipeline with two types of controllers – a PID controller and a Reinforcement Learning (RL) controller – for the Opentrons OT-2 robotic pipette. The goal was to enable the robotic pipette to inoculate root tips present in multiple specimen plates. This required adapting the controllers to accept the output of the computer vision pipeline, which involved ensuring that the root tip coordinates identified by the vision system were appropriately formatted and relayed to the controllers. Unfortunately, I was unable to work on this task.

## Task 14: Performance Benchmarking

In this task, the focus was on benchmarking the performance of two control systems developed for the Opentrons OT-2 robotic pipette: a PID controller and a Reinforcement Learning (RL) controller. The key objective was to quantify and compare their performance using relevant metrics, primarily focusing on speed and accuracy. This involved devising effective methods to measure these metrics and establishing a comparative framework to evaluate the controllers' efficiency.

The benchmarking process was crucial to understand how each controller performed under similar conditions and to identify which system was more effective for the specified tasks with the robotic pipette. A systematic approach was needed to ensure that the performance metrics were measured consistently and accurately for both controllers.

However, I was unable to carry out this task. The process of benchmarking these systems required a detailed understanding and implementation of measurement techniques, which I could not complete within the given timeframe.

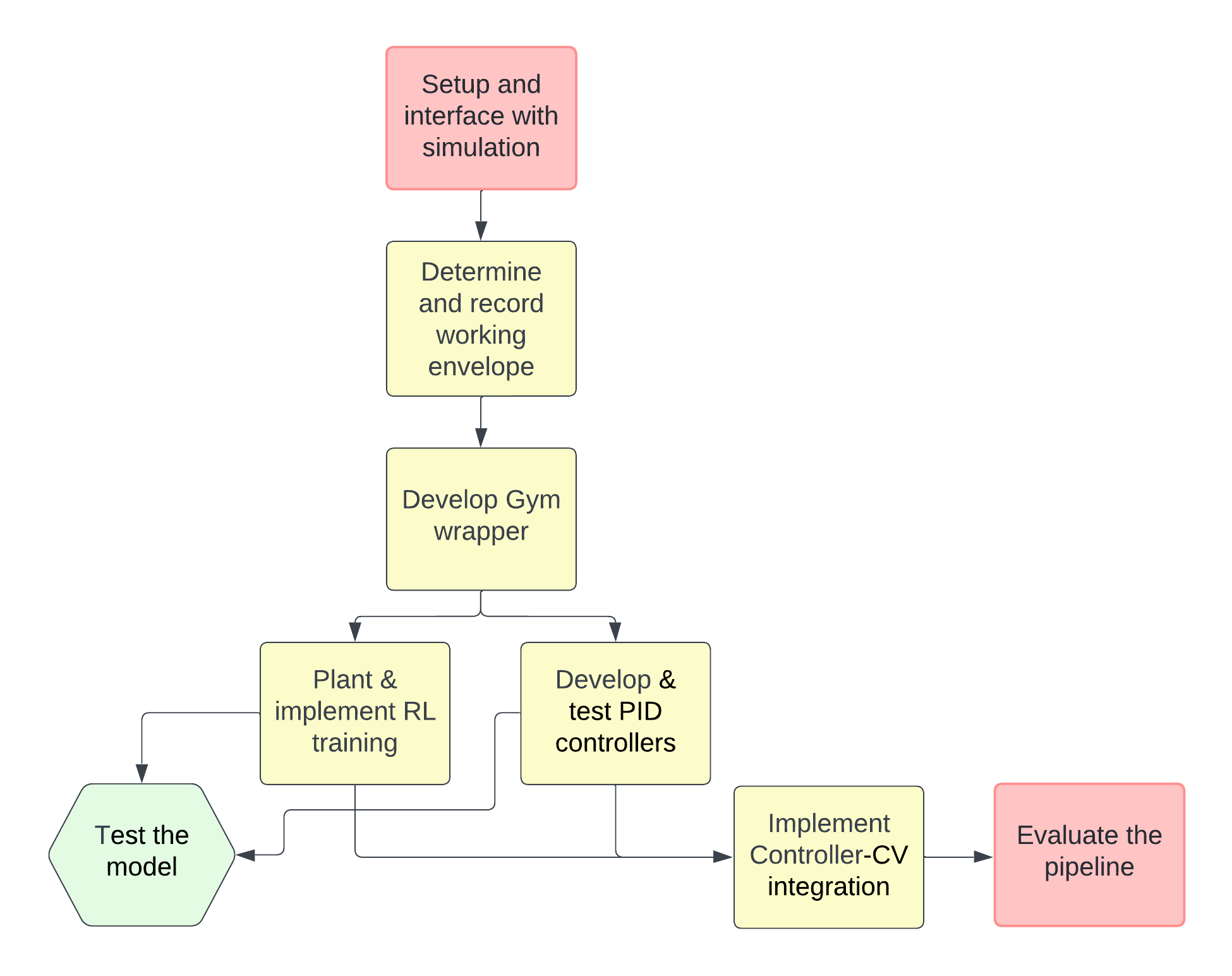


Figure : Reinforcement Learning (RL) pipeline flowchart

The development pipeline for the robotic control system begins with establishing a simulation environment. This environment is critical for testing and development as it provides a virtual platform that emulates the physical attributes and capabilities of the Opentrons OT-2 robot. To interact with this environment, a two-way communication interface is set up, which allows for the sending of robot commands and the receipt of observations regarding the robot's state.

Following this, the precise working envelope of the robot's pipette tip is ascertained. This involves maneuvering the pipette to the extremities of the cubic space that it can operate within and recording the coordinates at each of these points. There are eight such points, corresponding to the corners of the cube, which together define the limits of the robot's range of motion.

The next phase in the pipeline involves the development of a Gym wrapper. This wrapper serves as an adaptation layer that reformats the existing simulation environment to conform to the Gym interface, which is a standard in the reinforcement learning community. This interface specification includes defining the action and observation spaces, which encapsulate the possible actions that the robot can take and the potential observations or states that it can encounter.

Parallel to the creation of the Gym wrapper, a process is established for training a reinforcement learning model. This model utilizes the Gym environment to learn optimal strategies for moving the pipette tip to any designated position within the working envelope. Training a reinforcement learning model is a complex task that involves experimenting with various algorithms and hyperparameters to discover the most effective combination for the given application.

Concurrently, a PID controller is developed and tested. This controller is designed to provide precise control over the robot's movements, with separate PID controllers for each axis of motion. Tuning the PID controllers involves adjusting their gain parameters to achieve a balance between responsiveness and stability in the robot's movements.

Upon completion of the Gym wrapper, reinforcement learning training, and PID controller development, the pipeline advances to the integration of the controller with a computer vision pipeline. This integration is essential for tasks where the robot's movements must be guided by visual input, such as locating and acting upon specific points of interest in the robot's environment. The output from the computer vision pipeline, typically coordinates of target locations, must be converted into a format that the controller can interpret to guide the robot's movements.

The final stage of the pipeline is the evaluation of the entire system. This involves testing the robot's performance in carrying out its designated tasks, which in this context, include inoculating root tips present in multiple specimen plates. The evaluation assesses the system's accuracy, speed, and reliability, ensuring that it meets the requirements of the application.

Conclusion

This project, focusing on the integration of computer vision and robotics for plant science research, particularly Arabidopsis roots, has demonstrated significant achievements. The development of a sophisticated image processing pipeline and the application of machine learning techniques have been pivotal in automating and enhancing research methodologies in plant science. These technical advancements have not only provided practical solutions but have also paved the way for future research and development in the field.

The learning experience gained from this project has been multifaceted. The complexities encountered in dataset management, algorithm refinement, and integration of interdisciplinary knowledge have enriched our understanding of both the potential and challenges in applying advanced technology to scientific research. This process emphasized the importance of adaptability, continuous learning, and the ability to work collaboratively across different fields.

Moreover, the project has been instrumental in honing key skills such as critical thinking, problem-solving, and effective communication. The challenges faced and overcome have provided valuable lessons in perseverance and innovation. The experience has underscored the importance of a systematic approach to research and development, ensuring that every step, from conception to execution, is well-planned and executed.

In conclusion, this project has not only met its technical objectives but also contributed significantly to personal and professional growth. The insights and skills developed during this endeavor will undoubtedly be valuable in future scientific and technological pursuits.

References

Noyan, M. A. (2022). Uncovering bias in the PlantVillage dataset. *arXiv preprint arXiv:2206.04374*.

[[2206.04374] Uncovering bias in the PlantVillage dataset (arxiv.org)](https://arxiv.org/abs/2206.04374)

Torralba, A., & Efros, A. A. (2011, June). Unbiased look at dataset bias. In *CVPR 2011* (pp. 1521-1528). IEEE.

[Unbiased Look at Dataset Bias (mit.edu)](https://people.csail.mit.edu/torralba/research/bias/)

*ChatGPT*. (n.d.). <https://chat.openai.com/>

*Used for rewriting the text in this document.*

1. Torralba, A., & Efros, A. A. (2011, June). Unbiased look at dataset bias. In *CVPR 2011* (pp. 1521-1528). IEEE.

   [Unbiased Look at Dataset Bias (mit.edu)](https://people.csail.mit.edu/torralba/research/bias/) [↑](#footnote-ref-1)
2. Noyan, M. A. (2022). Uncovering bias in the PlantVillage dataset. *arXiv preprint arXiv:2206.04374*.

   [[2206.04374] Uncovering bias in the PlantVillage dataset (arxiv.org)](https://arxiv.org/abs/2206.04374) [↑](#footnote-ref-2)