VISION-BASED ROBOTIC PLANT GROWTH MONITORING AND LEARNING IN POLYCULTURE FARMING

by

Arif Yılmaz

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2021

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2024

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor, Prof. H. Işıl Bozma, for her encouragement, great ideas, and supportive attitude.

I would like to thank Prof. Yağmur Denizhan and Assoc. Prof. Mehmet Metin Ozgüven for their participation in my thesis committee.

Special thanks and love to Yavuzhan Erdem, Serhat İşcan, Süleyman Batuhan Vatan, Semanur Kumaş, Ahmet Harun Eker, and all ISL members for their support.

Finally, my heartfelt thanks go to my family and friends for their support, love, and understanding throughout this journey. I am especially grateful to my brother, mother, and father, whose patience and encouragement during difficult periods were invaluable. I appreciate Berkay Oğuz Aykan and Hakan Karataş, who stood by me throughout the thesis process, and my cats, whose comforting presence was a constant source of solace. A special mention goes to İrem Gülçin Zırhlıoğlu, whose readiness to share her experiences and offer help, even from a distance, was truly heartening.

This work has been supported in part by ROYAL CB SBB 2019K12-149250.

ABSTRACT

VISION-BASED ROBOTIC PLANT GROWTH MONITORING AND LEARNING IN POLYCULTURE FARMING

This thesis is focused on vision-based robotic plant growth monitoring and learning in polyculture farming. This type of agriculture dictates special soil preparation and farming a variety of companion plants together and hence differs from monoculture agriculture. The three-axis cartesian robot TarRob endowed with vision is used. First, its workspace is set up as an indoor polyculture garden through preparing the soil and planting a variety of companion plants. The behavioral capabilities of TarRob are extended to collect RGB-D images of these plants regularly. A polyculture plant dataset is initiated by adapting a four times daily schedule six hours apart over three months. Next, learning-based models for polyculture plant detection, plant segmentation, and leaf detection have developed. These models are primarily YOLO-based models that are subjected to further training. For this, the dataset is expanded as to include ground truth segment and leaf data. Following, the computation of the structural phenotypes is considered. A three-dimensional model of each plant is constructed via combining RGB-based detection and segmentation results with the depth data and is then used to compute the major structural phenotypes including height, width, length, area, and leaf count. Finally, the temporal nature of these phenotypes is studied from the resulting time series data. In particular, various statistical methods and learning models can be used to learn how each plant type grows. This thesis presents a comprehensive, fast, and modular approach to the automated monitoring and analysis of plant growth in polyculture farming by integrating robotics with machine vision.

ÖZET

POLİKÜLTÜR TARIMDA BİTKİ GELİŞİMLERİNİN ROBOTLA GÖRSEL İZLENMESİ VE ÖĞRENİMİ

Bu tez çalışması, polikültür tarımda bir robot ile görüntü tabanlı bitki büyümesi izleme ve öğrenme problemlerine odaklanmaktadır. Polikültür tarım, toprağın hazırlanması ve bahçe birliklerinin oluşturulması gibi prensipleriyle monokültür tarımdan oldukça farklı bir tarım türüdür. Çalışmada daha önce geliştirilmiş olan üç eksenli kartezyen robot TarRob, ek geliştirmeler yapılarak kullanılmaktadır. İlk önce, robotun çalışma alanı olan bahçenin toprağı polikültür ekim prensiplerine uygun şekilde hazırlanmış ve birlik oluşturabilecek bir bitki seti seçilerek dikilmiştir. TarRob'un görev seti tüm ekili bitkilerin RGB-D görüntülerini düzenli toplama işi yapabilecek şekilde güncellenmiştir. Üç ay süresince, günde altı saat aralıkla dört kez veri toplanarak polikültür bitki veri seti oluşturulmuştur. İkinci kısımda, robotun kendi başına ve bitki türünden bağımsız olarak bitkileri ve yaprakları tespit edebilmesine yönelik modeller geliştirilmiştir. Bu modeller esasen RGB verisi kullanan YOLO derin ağları temelli olup, ek bir eğitime tabi tutulmuşlardır. Bunun için, veriseti ayrıca doğru bölütleme sonuçları içerecek şekilde genişletilmiştir. Üçüncü kısımda, RGB tabanlı kabataslak ve ince bölütleme sonuçları, derinlik verileriyle birleştirilerek her bir bitkiye ait nokta bulutunu oluşturan metodlar geliştirilmiştir. Nokta bulutu verisi kullanılarak, yükseklik, genişlik, uzunluk, alan ve yaprak sayısı gibi bitkinin yapısal özellikleri otomatik olarak hesaplayan metodlar geliştirilmiştir. En son kısmında ise, bu özelliklerin zamansal değişimleri kullanılarak bitkilerin büyüme modellerinin öğrenilmesine yönelik çalışılmalar yapılmıştır. Bu çerçevede tez, robotik, makine görme ve makine öğrenmesi alanlarını birleştirerek, polikültür tarımda bitki büyümesinin otomatik olarak izlenmesi ve analizi için kapsamlı, hızlı ve modüler bir yaklaşım sunmaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF SYMBOLS . . . xiv
LIST OF ACRONYMS/ABBREVIATIONS . . . 1
1. INTRODUCTION . . . 2
1.1. Proposed Approach . . . 4
1.2. Contributions of the Thesis . . . 5
1.3. Thesis Outline . . . 5
2. POLYCULTURE GARDEN & TARROB SETUP . . . 7
2.1. Soil Preparation . . . 7
2.2. Plant Selection . . . 9
2.3. Garden Layout and Plant Seeding . . . 9
2.4. TarRob Plant Dataset . . . 11
3. VISUAL PLANT DETECTION & SEGMENTATION . . . 21
3.1. Related Literature . . . 21
3.2. RGB-based Plant Detection & Segmentation . . . 22
3.2.1. Learning Plant Detection and Segmentation Models . . . 22
3.2.2. Plant Tracking . . . 25
3.3. RGB-based Leaf Detection . . . 27
3.4. TarRob Plant Detection & Segmentation Dataset . . . 31
3.4.1. Dataset Annotation . . . 32
3.4.2. Data Augmentation . . . 34
3.4.3. Dataset Cleaning . . . 34
3.5. Experimental Results . . . 37
3.5.1. Plant Detection & Segmentation . . . 37

3.5.2. Leaf Detection 44
3.5.3. Evolution of Plant & Leaf Detection 52
4. RGB-D BASED COMPUTATION OF STRUCTURAL PHENOTYPES . . 54
4.1. Related Literature 54
4.2. Structural Phenotypes 55
4.3. Structural Phenotype Computation Pipeline 56
4.4. Experimental Results 57
4.4.1. Depth Data 57
4.4.2. Accuracy of Geometric Computations 58
4.4.3. Structural Phenotype Measurements 60
4.4.4. Structural Phenotype Measurements on Various Plant Species . 62
5. LEARNING AND PREDICTING PLANT GROWTH 68
5.1. Related Literature 68
5.2. Evolution of Structural Phenotypes 69
5.3. Statistical Forecasting on Time-Series 78
5.3.1. Estimating ARIMA Parameters 78
5.3.2. Testing ARIMA Predictions 79
6. CONCLUSION 83
REFERENCES 85
APPENDIX A: ROBOT HARDWARE 93
A.1. TarRob Hardware 93
A.2. Intrinsic Camera Parameters 94
A.3. Lens Distortion Coefficients 94
APPENDIX B: SOFTWARE DETAILS 96
B.1. Gazebo Simulation of the TarRob 96
B.2. Plant Growth Monitoring System 96
B.3. Blender-Gazebo Connection 99
APPENDIX C: MODEL TRAINING DATASETS 100

LIST OF FIGURES

Figure 1.1. TarRob robot and its gardening facility. 3

Figure 1.2. Proposed approach for visual plant monitoring and learning. . . . 4

Figure 2.1. Soil preparation. 8

Figure 2.2. Polyculture garden design [20]. 9

Figure 2.3. Polyculture garden layout. 11

Figure 2.4. Garden layout after planted seedlings. 12

Figure 2.5. RGB data of lettuce plants vs time. 14

Figure 2.6. RGB data of lettuce plants vs time (cont.). 15

Figure 2.7. RGB data of chard plants vs time. 16

Figure 2.8. RGB data of chard plants vs time (cont.). 17

Figure 2.9. RGB data of spring onion plants vs time. 18

Figure 2.10. RGB data of spring onion plants vs time (cont.). 19

Figure 2.11. RGB data of broccoli plants vs time. 20

Figure 3.1. YOLOv8 plant detection-segmentation model training results. . . 24

Figure 3.2. Plant track algorithm result. 26

Figure 3.3. YOLOv7 leaf detection model training results. 28

Figure 3.4. YOLOv8 leaf detection model training results. 29

Figure 3.5. Fine-tuned YOLOv8 leaf detection model training results. 30

Figure 3.6. Leaf detection examples of the trained models. 31

Figure 3.7. Data annotation pipeline. 33

Figure 3.8. Data annotation pipeline on an example broccoli image. 33

Figure 3.9. Augmentation example. 34

Figure 3.10. Sample RGB data with localization problem. 35

Figure 3.11. Sample RGB image with a lighting problem. 36

Figure 3.12. Detection and segmentation of lettuces vs time. 38

Figure 3.13. Detection and segmentation of lettuces vs time (cont.). 39

Figure 3.14. Detection and segmentation of chards vs time. 40

Figure 3.15. Detection and segmentation of chards vs time (cont.). 41

Figure 3.16. Detection and segmentation of spring onions vs time. 42

Figure 3.17. Detection and segmentation of spring onions vs time (cont.). . . . 43

Figure 3.18. Detection and segmentation of broccolis vs time. 44

Figure 3.19. Detection of lettuce leaves vs time. 45

Figure 3.20. Detection of lettuce leaves vs time (cont.). 46

Figure 3.21. Detection of chard leaves vs time. 47

Figure 3.22. Detection of chard leaves vs time (cont.). 48

Figure 3.23. Detection of spring onion leaves vs time. 49

Figure 3.24. Detection of spring onion leaves vs time (cont.). 50

Figure 3.25. Detection of broccoli leaves vs time. 51

Figure 3.26. Evolution of plant and leaf detection 53

Figure 4.1. Overall structural phenotype computation pipeline. 56

Figure 4.2. Wooden blocks setup 58

Figure 4.3. Width and length measurement of the cardboard box. 60

Figure 4.4. Measurement verification for cardboard box. 61

Figure 4.5. Measurement verification for the whole system. 61

Figure 4.6. Structural phenotype computation pipeline stages on pepper image. 63

Figure 4.7. Structural phenotype computation pipeline stages on tomatoes image. 65

Figure 4.8. Structural phenotype computation pipeline stages on beet image. . 66

Figure 5.1. Temporal evolution of structural phenotypes - lettuces L_1 and L_2. 70

Figure 5.2. Temporal evolution of structural phenotypes - lettuces L_3 and L_4. 71

Figure 5.3. Temporal evolution of structural phenotypes - lettuces L_5. 72

Figure 5.4. Temporal evolution of structural phenotypes - chards C_1 and C_2. . 73

Figure 5.5. Temporal evolution of structural phenotypes - chards C_3 and C_4. . 74

Figure 5.6. Temporal evolution of structural phenotypes - chard plant C_5. . . 75

Figure 5.7. Temporal evolution of structural phenotypes - broccolis B_1 and B_2. 76

Figure 5.8. Temporal evolution of structural phenotypes - broccoli plant B_3. . 77

Figure 5.9. ARIMA prediction on structural phenotypes - broccoli B_2 and chard C_4. 80

Figure 5.10. ARIMA prediction on structural phenotypes - lettuce L_4. 81

Figure B.1. ROS network 98

LIST OF TABLES

Table 2.1. Plant growth conditions 10

Table 2.2. Planting distance and harvest time of the plants. 10

Table 2.3. Number of seedlings planted 11

Table 3.1. Training performance of the plant detection 23

Table 3.2. Training performance of the plant segmentation 24

Table 3.3. Fine-tuned YOLOv7 - Leaf detection training performance. 27

Table 3.4. Trained YOLOv8 - Leaf detection training performance. 29

Table 3.5. Fine-tuned YOLOv8 - Leaf detection training performance. 30

Table 4.1. Camera distance to stacked wooden blocks 59

Table 4.2. Height accuracy. 59

Table 4.3. Cardboard box scenario 60

Table 4.4. Leaf instance's dimensions. 62

Table 4.5. Dimensions of the pepper. 64

Table 4.6. Dimensions of the tomatoes. 66

Table 5.1. ARIMA parameters (p, d, q) for different plant features. 79

Table 5.2. NRMSE values for plant features across different plant species. . . 82

Table A.1. ZED 2 Camera intrinsic parameters. 94

Table A.2. ZED 2 Camera distortion coefficients. 95

Table C.1. Plant detection dataset 100

Table C.2. Leaf detection dataset 101

Table C.3. External leaf detection dataset. 101

LIST OF SYMBOLS

A_b	Bounding box area
A_p	Plant area as viewed orthographically
b_i	The i-th bounding box
B_i	The i-th broccoli from garden
$\mathcal{B}$	Set of bounding boxes
C_I	Image center coordinates
C_i	The i-th chard from garden
C_{b_i}	Center coordinates of the i-th bounding box
c_x, c_y	Coordinates of the image center
d	Differencing order in ARIMA model
d_i	Distance between box center and image center
$d_{n_{\text{stack}}}$	Computed depth for n_{stack} stacked blocks
$d_{n_{\text{stack}}+1}$	Computed depth for $n_{\text{stack}} + 1$ stacked blocks
$\mathcal{D}_E$	External leaf detection dataset
$\mathcal{D}_L$	Manually annotated leaf detection dataset
$\mathcal{D}_P$	Plant detection dataset
f_x, f_y	Focal lengths of the camera in x and y directions
h_{comp}	Computed height of a single block
i^*	Index of the most centered bounding box
k_1, k_2, k_3	Radial distortion coefficients of the camera
L_i	The i-th lettuce from garden
m_i	Array of computed metric values
n	Total number of observations
n_{stack}	Number of stacked blocks
N	Total number of bounding boxes
N_b	Number of pixels in the bounding box
N_s	Number of pixels in the segment
O_i	The i-th spring onion from garden

p	AutoRegressive order in ARIMA model
P	Precision
p_1, p_2	Tangential distortion coefficients of the camera
q	Moving Average order in ARIMA model
R	Recall
W	Weight vector for fitness score calculation
$x_1^i, y_1^i, x_2^i, y_2^i$	Corner coordinates of the i-th bounding box
y_i	Observed value of the plant feature at time i
$\hat{y}_i$	Predicted value of the plant feature at time i
$\bar{y}$	Mean of the observed values

LIST OF ACRONYMS/ABBREVIATIONS

1D CNN	One Dimensional Convolutional Neural Network
2D	Two Dimensional
3D	Three Dimensional
ANN	Artificial Neural Network
AP	Average Precision
ARIMA	AutoRegressive Integrated Moving Average
BIC	Bayesian Information Criterion
CNN	Convolutional Neural Network
Conv-LSTM	Convolutional Long Short-Term Memory
CV	Computer Vision
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
IoU	Intersection over Union
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
Mask R-CNN	Mask Region-based Convolutional Neural Network
ML	Machine Learning
NRMSE	Normalized Root Mean Square Error
PCL	Point Cloud
RGB	Red Green Blue
RGB-D	Red Green Blue and Depth
RNN	Recurrent Neural Network
ROLO	Recurrent YOLO
ROS	Robot Operating System
SAM	Segment Anything Model

1. INTRODUCTION

The need for sustainable agricultural practices that ensure food security and protect the environment is becoming even more critical with the increasing global population and expanding urban regions. In this aspect, polyculture farming and agricultural automation have become very important.

Polyculture farming is an important study area due to its potential benefits. Polyculture farming utilizes biodiversity - inspired by the dynamics of natural ecosystems [1,2]. As such, while it enables the diversity of crops, soil preservation is improved as it minimizes pesticide reliance. This is indeed a big problem since the 1950s, fifty percent of soil has been lost, with predictions of further depletion. This issue, affecting both developed and developing countries, is even worsened by the overuse of groundwater and biodiversity loss [3].

However, polyculture demands even more intensive labor than traditional monoculture agriculture. Recent developments in agricultural automation promise reduced human intervention and lower expenses [4–7]. This trend also presents opportunities for technological advancements in polyculture methodologies. Machine vision and deep learning applications show potential for plant trait estimation and growth monitoring in agricultural systems [8–11]. Machine vision technology in agricultural automation [12] and recent comparisons of automated systems with traditional methods [13] demonstrate the potential for reducing labor needs and resource consumption.



Figure 1.1. TarRob robot and its gardening facility.

This thesis is focused on plant growth monitoring and polyculture farming with a gantry-type robot endowed with vision. In particular, garden-based farming automation is considered. For this, TarRob as seen in Figure 1.1 is used. This robot has been motivated by the FarmBot project [14] and has been developed through a series of senior projects [15–17]. The robot moves over a garden of size 2 meters by 0.9 meters by 0.4 meters. Its motion mechanism consists of three prismatic degrees of freedom. Its workspace is rectangular and shaped to size 2 meters by 0.9 meters by 0.4 meters. This area corresponds to the robot's gardening facility. TarRob is endowed with a stereo camera that provides RGB-D data. The robot can collect RGB-D data daily according to a pre-determined schedule - i.e. every 6 hours.

Next, the robot's visual monitoring and learning capabilities need to be developed. This development is crucial for accurately measuring the three-dimensional growth of plants. In particular, we are interested in non-invasive methods since otherwise future plant growth may be adversely affected. Also, the manual nature of such measurements can be quite time-consuming, tedious, and error-prone. While there are some recently reported efforts in this direction, these have their limitations. It should be noted that although the garden is a controlled environment with the positions of the plants being known, both segmentation of individual plants and segmentation of leaves turn out to be challenging due to the polyculture farming and varying tropic movements of individual plants. Furthermore, as plants grow, leaves from different plants tend to

become overlapping so distinguishing individual plants becomes difficult. In this aspect, the goal of the thesis is to develop an autonomous robotic system with vision-based plant monitoring and learning for polyculture farming.

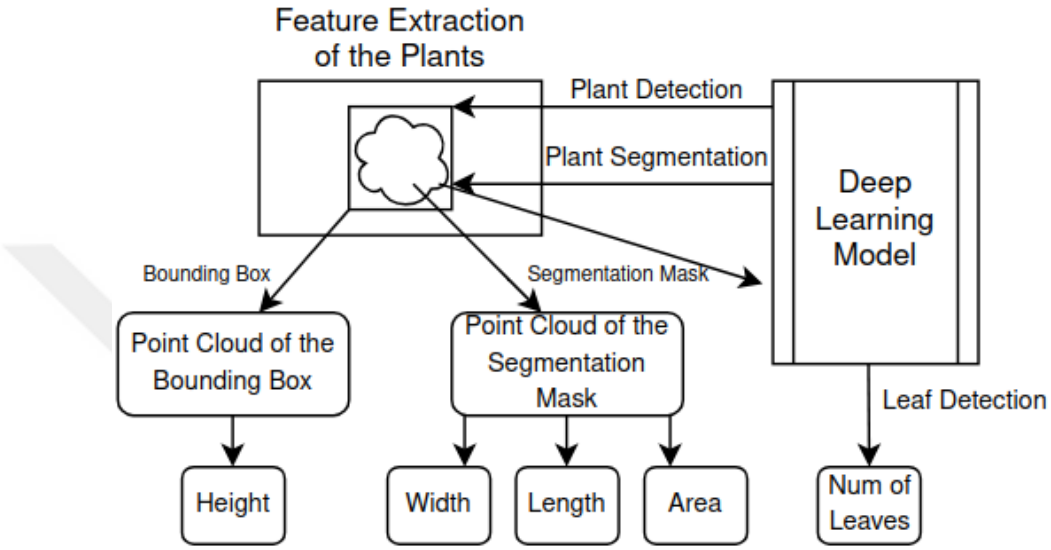


Figure 1.2. Proposed approach for visual plant monitoring and learning.

1.1. Proposed Approach

First, a polyculture garden is set up and a set of companion plants are selected. These plants are then planted. The behavioral capabilities of TarRob have been extended to collect RGB-D images of these plants regularly. A polyculture plant dataset is created by adapting a four-times daily schedule six hours apart over three months. Following, visual monitoring and learning problems are considered. The proposed approach is as shown in Figure 1.2. First, each plant is segmented through finding the corresponding bounding box and then refining it to determine its contour segment. For this, RGB data is used and an existing convolutional neural network (CNN) is trained. Following, a leaf detector network is trained in order to segment the leaves of each detected plant - again using RGB data. For this, the RGB-D dataset is further expanded to include ground truth segment knowledge. Following, RGB-based detection and segmentation results are combined with the corresponding depth data to estimate the structural phenotypes of each detected plant - including height, width, length, area,

and number of leaves. Their values are recorded daily to track each plant's growth. The resulting temporal data can then be used to learn the growth model of each plant species. As one can see in the flowchart, new models to be trained can easily be added to the network, or a new feature measurement method can be added to the system.

1.2. Contributions of the Thesis

The contributions of this thesis can be summarized as follows:

(i) TarRob Dataset: A novel RGB-D polyculture dataset featuring broccoli, lettuce, chard, and spring onion plants is constructed. This dataset captures plant growth stages in a controlled indoor environment and provides a resource for robotics research focused on polyculture farming.

(ii) Structural Phenotype Computation Pipeline: An integrated pipeline that combines YOLO-based detection and segmentation with depth information is proposed. The resulting pipeline is shown to accurately estimate plant height, width, length, area, and leaf count in a polyculture garden. This modular approach allows for easy integration of new detection or measurement methods.

(iii) Plant growth modeling and prediction: A method for modeling plant growth is introduced. The model is based on the time-series data of structural phenotypes and can be used in monitoring plant growth.

1.3. Thesis Outline

The outline of the thesis is as follows:

- In Chapter 2, the polyculture garden and TarRob setup are described. This includes details of soil preparation, plant selection, garden layout and TarRob dataset.
- Chapter 3 focuses on visual plant detection. Related literature is reviewed. Following, RGB-based plant detection and segmentation and leaf detection are ex-

plained. Experimental results on the TarRob dataset are discussed.

- In Chapter 4, RGB-D based computation of structural phenotypes is presented. First, the computed structural phenotypes and the details of their computation are explained. This is followed by experimental results on the accuracy of depth, geometric measurements and structural phenotype computations.
- Chapter 5 addresses learning and predicting plant growth. After reviewing related literature, plant growth data visualization and time-series prediction models on plant growth data are presented along with an experimental evaluation.
- Chapter 6 concludes the thesis with a summary of the work and potential areas for future research.
- Appendix A presents the technical details of the TarRob robot.
- In Appendix B, the details of the plant growth monitoring system and Gazebo simulation are provided.
- The datasets used in the training of the YOLO models are described in Appendix C.

2. POLYCULTURE GARDEN & TARROB SETUP

An indoor polyculture garden was set up using the gardening facility of the indoor farming robot TarRob. The following are required:

- Soil preparation
- Selection of companion plants
- Designing garden layout
- Planting

This work uses the canopy-level RGB-D data obtained by TarRobot - even though it is capable of getting data from other views as well. Note that high-throughput phenotyping can be achieved at the canopy-level sensory data when coupled with automatic or mobilized data acquisition [18]. For this, TarRob's behavioral capabilities have been extended to collect RGB-D images of these plants regularly. Furthermore, differing from most previous work, RGB data is combined with depth data, and thus point cloud models of plant instances can be constructed. Hence, monitoring is based on the analysis of three-dimensional information. The construction of the polyculture plant dataset is initiated by adapting a four-times daily schedule six hours apart over three months.

2.1. Soil Preparation

The first stage of creating a polyculture garden is the preparation of the soil. When preparing a proper polyculture garden, having organic and nutritious soil is as important as the diversity and companionship of the plants [19]. For this purpose, lime, manure, shrubs, organic compost, burnt animal manure, cardboard, and wood pieces were added to the soil content.

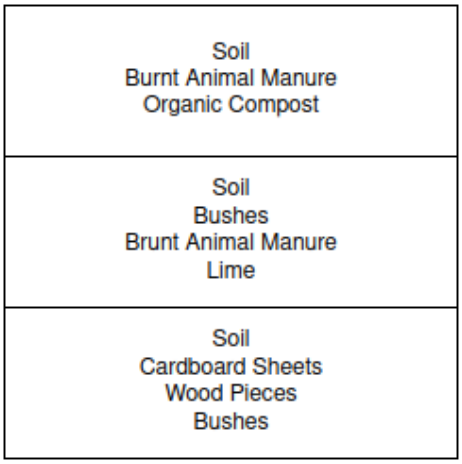


Figure 2.1. Soil preparation.

Figure 2.1 shows the layers of prepared soil content. Wood pieces, bushes, and cardboard sheets are in the bottom layer to enrich the soil over time, while burnt animal manure and organic compost are in the upper layer where plants can reach them to grow.

2.2. Plant Selection

The plants to be seeded in the polyculture garden should be companion plants and support each other's growth.

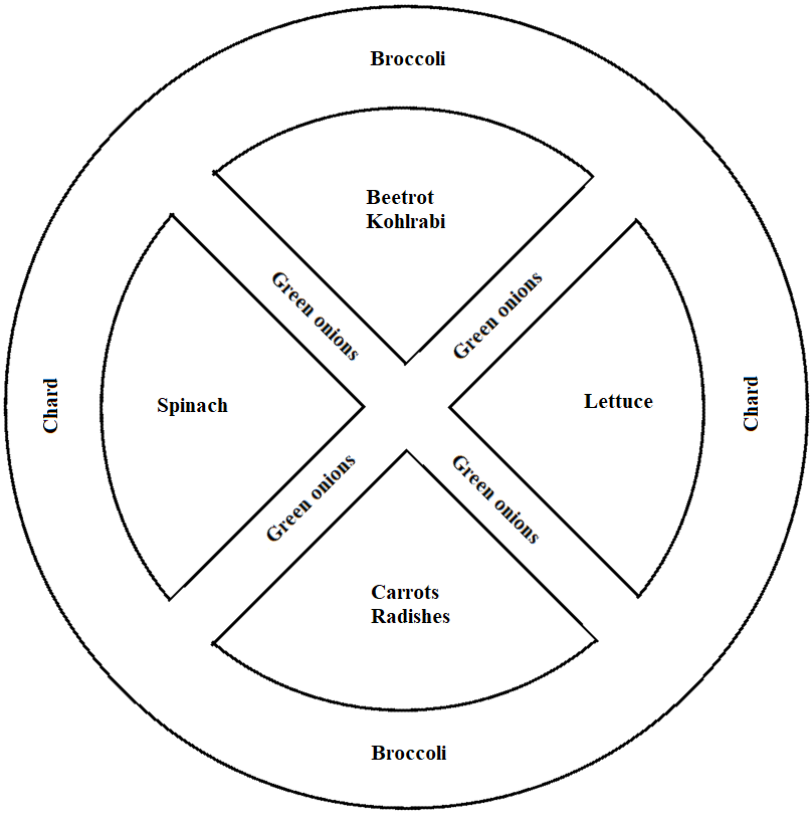


Figure 2.2. Polyculture garden design [20].

Figure 2.2 shows a polyculture garden design. Some of the selected plants from this circular garden design were chosen for our indoor polyculture garden [20]. These are broccoli, red beet, lettuce, spring onion, carrots, spinach, and chard.

2.3. Garden Layout and Plant Seeding

The next step in designing the polyculture garden is to provide the environmental conditions that are required by the selected plants. Ideal conditions for the selected plant species are given in Table 2.1. It is observed that a constant temperature of 18 degrees Celsius is a suitable temperature for all the plants. In addition, the required

lighting duration and soil pH values are also agreeable with each other [21–24].

Table 2.1. Plant growth conditions.

Plant	**pH**	**Temperature (°C)**	**Lighting Duration (hours)**
Broccoli	6.0-7.0	15-20	12-14
Lettuce	6.0	10-20	12-14
Carrot	5.5-6.5	16-18	12-16
Spring Onion	5.7-7.4	20-25	12-16
Garden Beet	6.0-7.0	16-18	12-16
Chard	6.0-8.0	10-23	12-14
Spinach	6.5-8.0	15-20	12-14

Table 2.2. Planting distance and growth time of the plants.

Plant	**Planting Distance (cm)**	**Growth Period (days)**
Broccoli	30-45	60-90
Lettuce	30-45	45-60
Carrot	7.5-10	60-80
Spring Onion	10-15	30-50
Garden Beet	10-20	50-70
Chard	20-25	50-70
Spinach	20-22	45-60

The layout was designed according to planting distance to preserve plant diversity and collect sufficient images from each plant species for the dataset. The ideal planting distances and growth period of the plants are as given in Table 2.2. While growth periods vary, this variance does not pose a problem. Once a plant is harvested, a seedling of the same plant species is planted in its place [21, 25]. The last stage before planting the plants is to turn the circular design into a rectangular area measuring 2 meters x 0.9 meters according to the planting distances. While choosing the areas shown in Figure 2.3, the relative plant positions according to each other were preserved.

Once the garden was set, plants were manually watered 2-3 times a week using a watering pot and making sure the soil was thoroughly moistened.

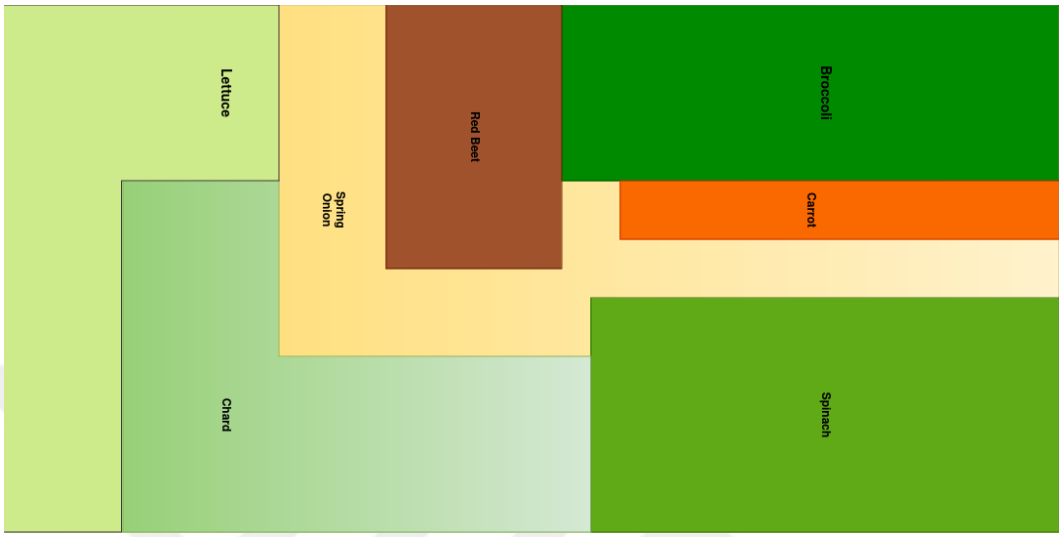


Figure 2.3. Polyculture garden layout.

2.4. TarRob Plant Dataset

Lettuce, chard, broccoli, and spring onion are planted as seedlings. The amount per species is given in Table 2.3. The remaining plants were sown as seeds. The number of plants sown as seeds is not exact. It should be remarked that planting seedlings has advantages since recommended plant distances can better be adhered to and data collection can be expedited.

Table 2.3. Number of seedlings planted for each plant species.

Plant Species	**Number of Seedlings**
Lettuce	5
Chard	5
Broccoli	3
Spring Onion	6

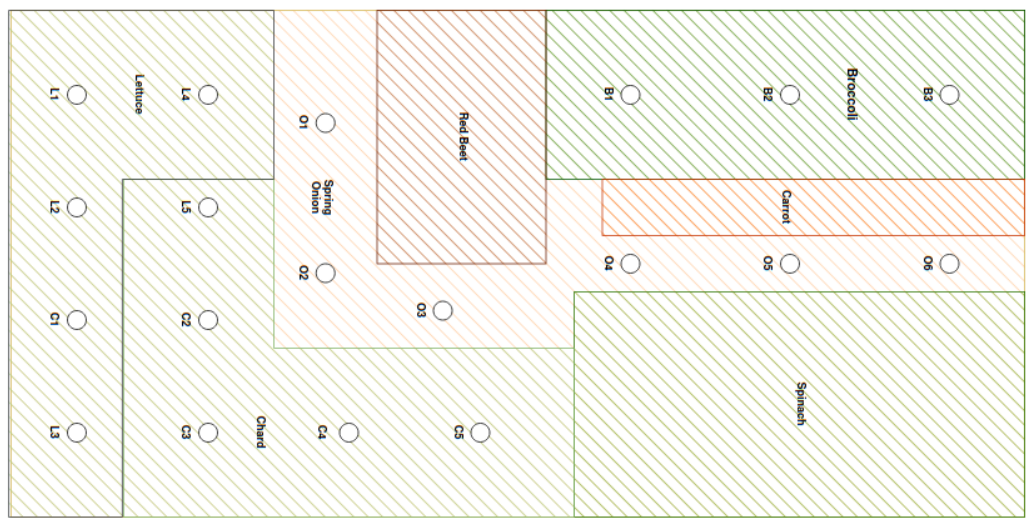


Figure 2.4. Garden layout after planted seedlings.

Figure 2.4 shows the position and naming convention of the planted seedlings. The first letter of the plant species is used as the first component of the naming. The second component refers to the plant id. Other IDs were given with an increasing order to each species.

The robot moves over the garden facility and takes RGB-D data using its ZED 2 camera. The home position of the robot is in the upper left corner of Figure 2.4. As the positions of the plants are known to it, it moves to these known positions in a horizontal scanning pattern. In particular, each data collection sequence starts with the plant L_1. After finishing the first row with L_3, the robot moves back to the L_4 and collects data of the plants located in the second row. It collects images of all plants with this horizontal scanning pattern. To ensure that the plants' light-dependent movements during different times of the day and non-directional nastic movements did not affect growth monitoring results, the established policy was to collect RGB-D data at six-hour intervals. Full spectrum lights were left on all day to support the light received by the indoor robot during the day and to enable taking images at night. Later in this system, the lights will be turned on at night only when the image is being taken to meet the darkness needs of the plant.

These images are then recorded in the robot's database. 1080p RGB images were saved with a .png extension. For the sake of resolution, depth images were saved with the .tiff extension to preserve depth information losslessly. Data was recorded as a 32-bit float using a CV bridge from the relevant ROS topic. To collect depth information accurately, the stereo camera needed to be at least 30 cm away from the plants.

The files were saved under folders named PlantID_PlantSpecies, with the format rgb/depth_yymmdd_hhmmss_x_y_z.png/tiff. For example, an RGB image of the second lettuce was saved as 2_Lettuce/rgb_20240217_175535_120_350_0.png. This data collection method ensures consistent and comprehensive monitoring of plant growth in the polyculture farm, capturing both visual and depth information at regular intervals. The first batch of mature lettuces was harvested and replaced with new seedlings on 24/04/2024 at 14:00.

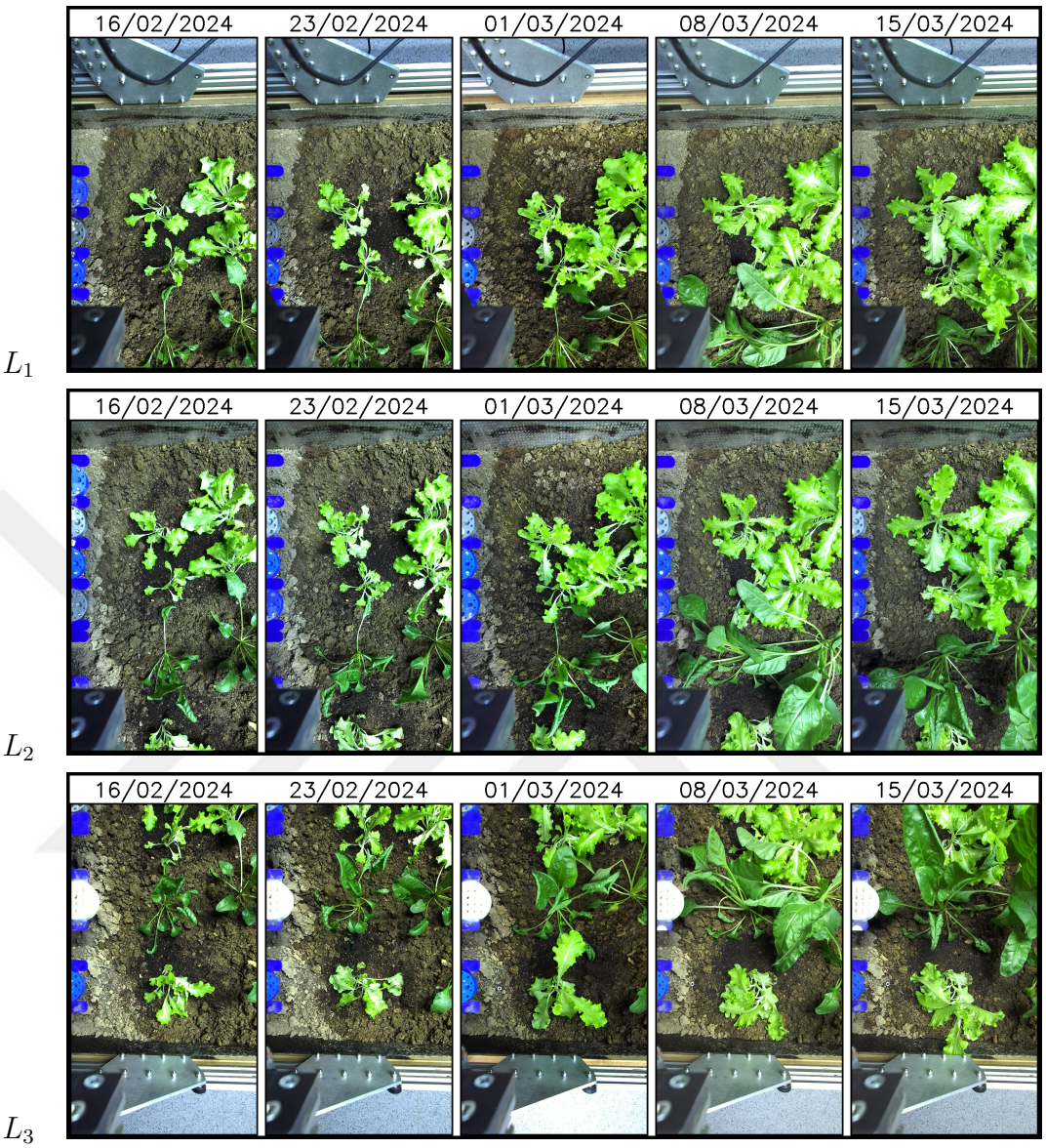


Figure 2.5. RGB data of lettuce plants vs time.

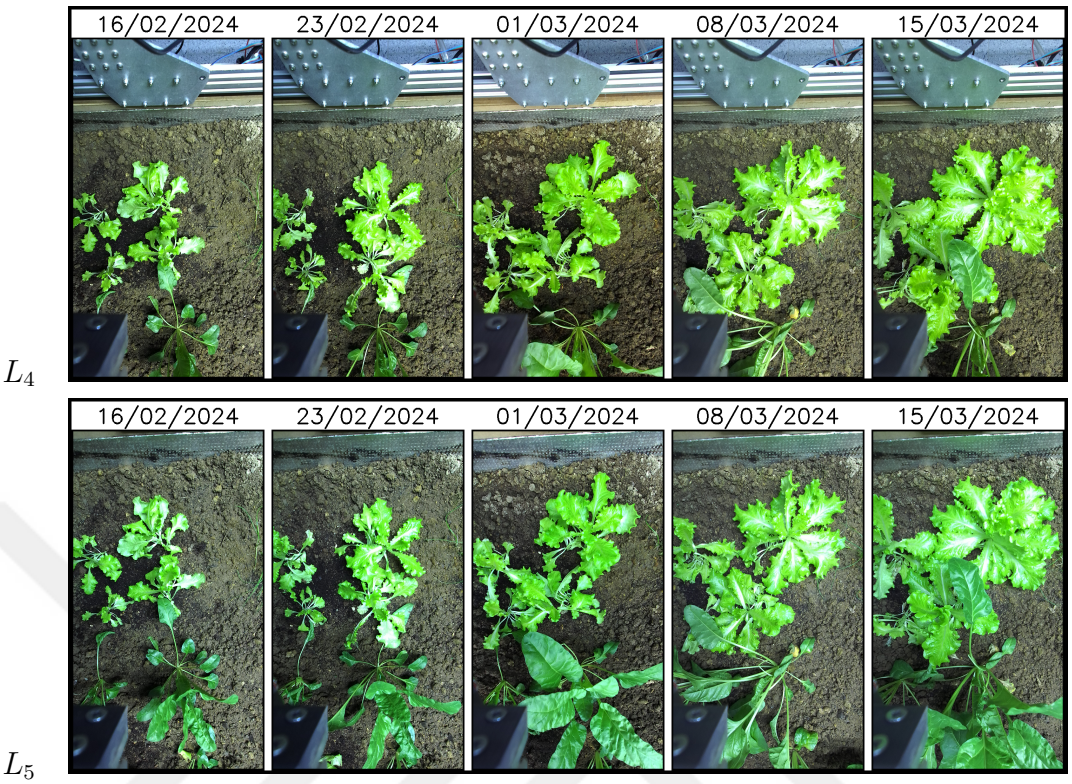


Figure 2.6. RGB data of lettuce plants vs time (cont.).

The evolution of the planted seedlings through roughly a five-week period at one-week intervals are shown separately for each plant species in Figure 2.5- Figure 2.11 show the evolution of the planted seedlings. Let it be noted that for the date of 01/03/2024, the robot was not correctly positioned while taking the data of some of these plants. For these plants, the data corresponding to slightly different dates (03/03/2024) are shown instead.

The development of lettuces is as seen in Figure 2.5, Figure 2.6. Planted L_1 and L_2 lettuces grew less growth than the rest. In particular, L_2 was negatively affected by the chard plant C_1. C_1 both occupied the space and prevented L_2 from receiving light. L_3 grew unhindered by the growth of other plants. In the indoor garden, lettuces should be planted on the sunny side.

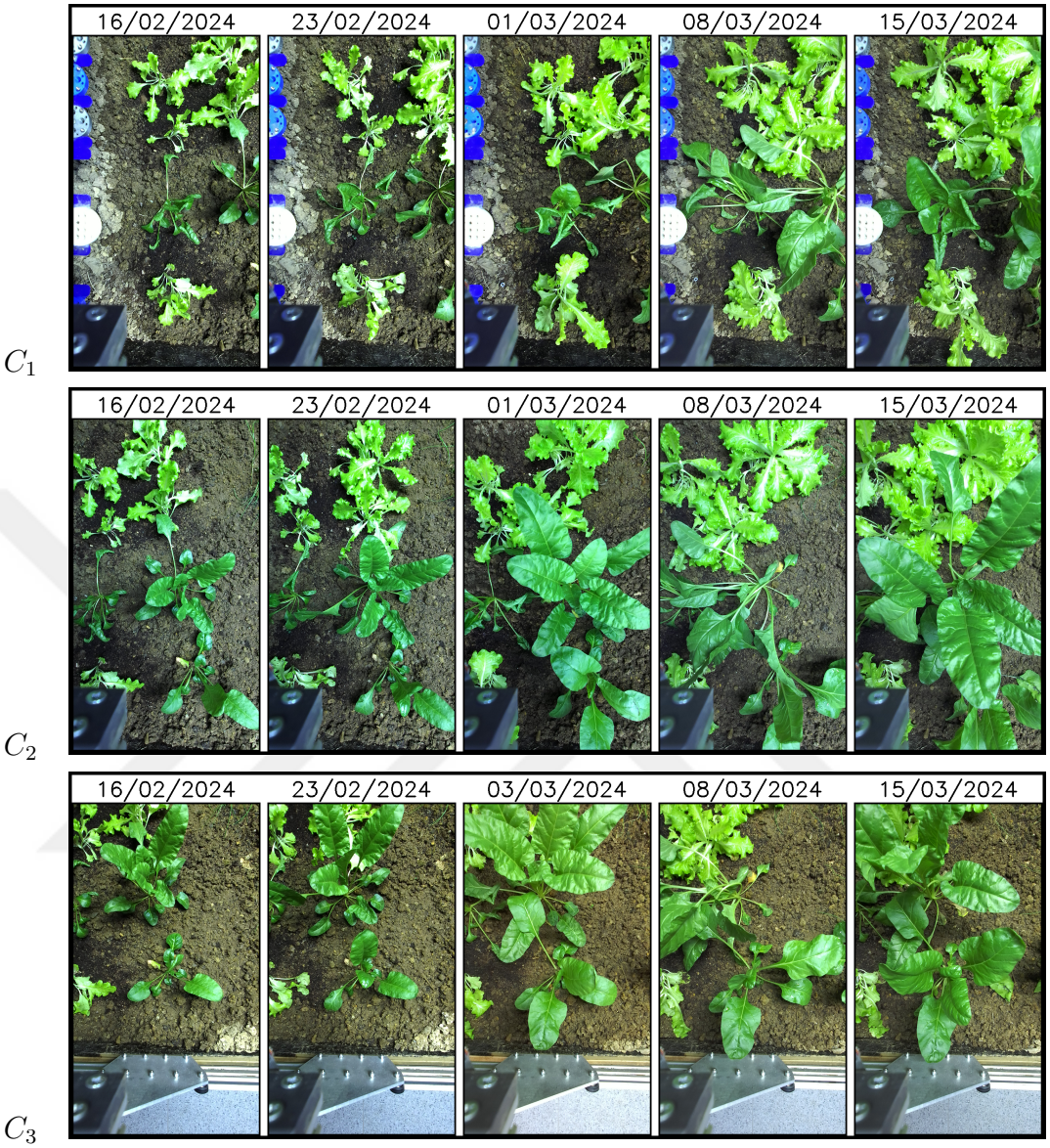


Figure 2.7. RGB data of chard plants vs time.

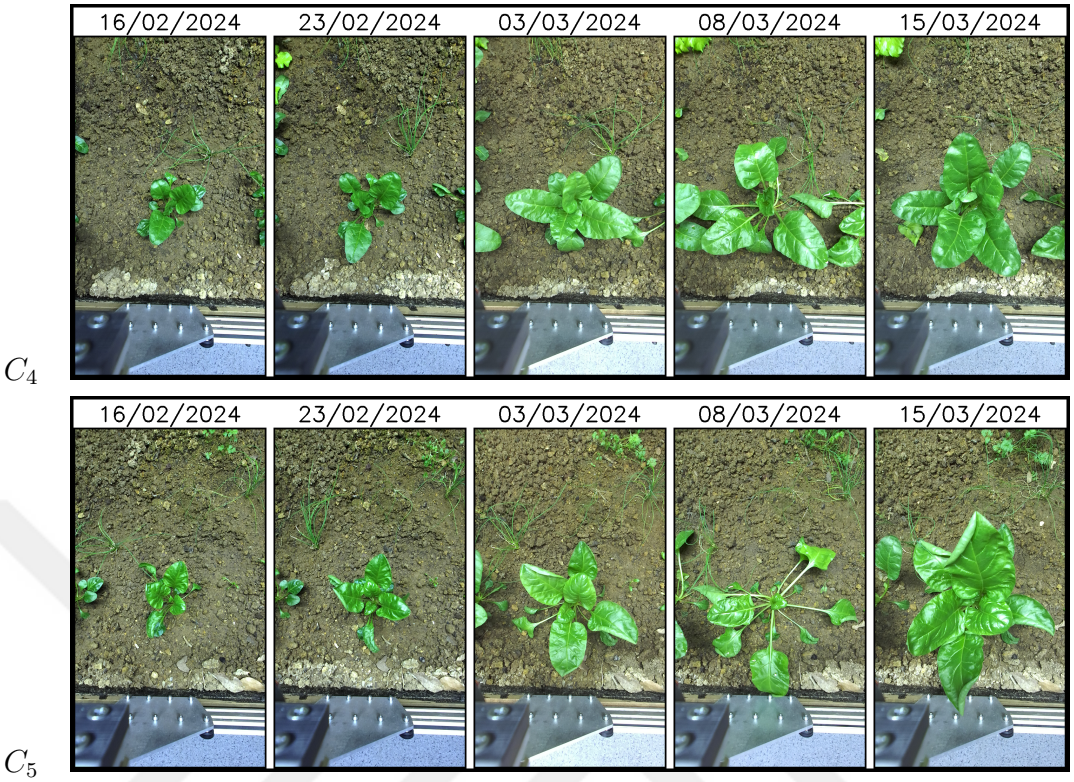


Figure 2.8. RGB data of chard plants vs time (cont.).

The progress of the chards is as seen in Figure 2.7 and Figure 2.8. All chards grew healthy. Their rapid growth and high height made them advantageous.

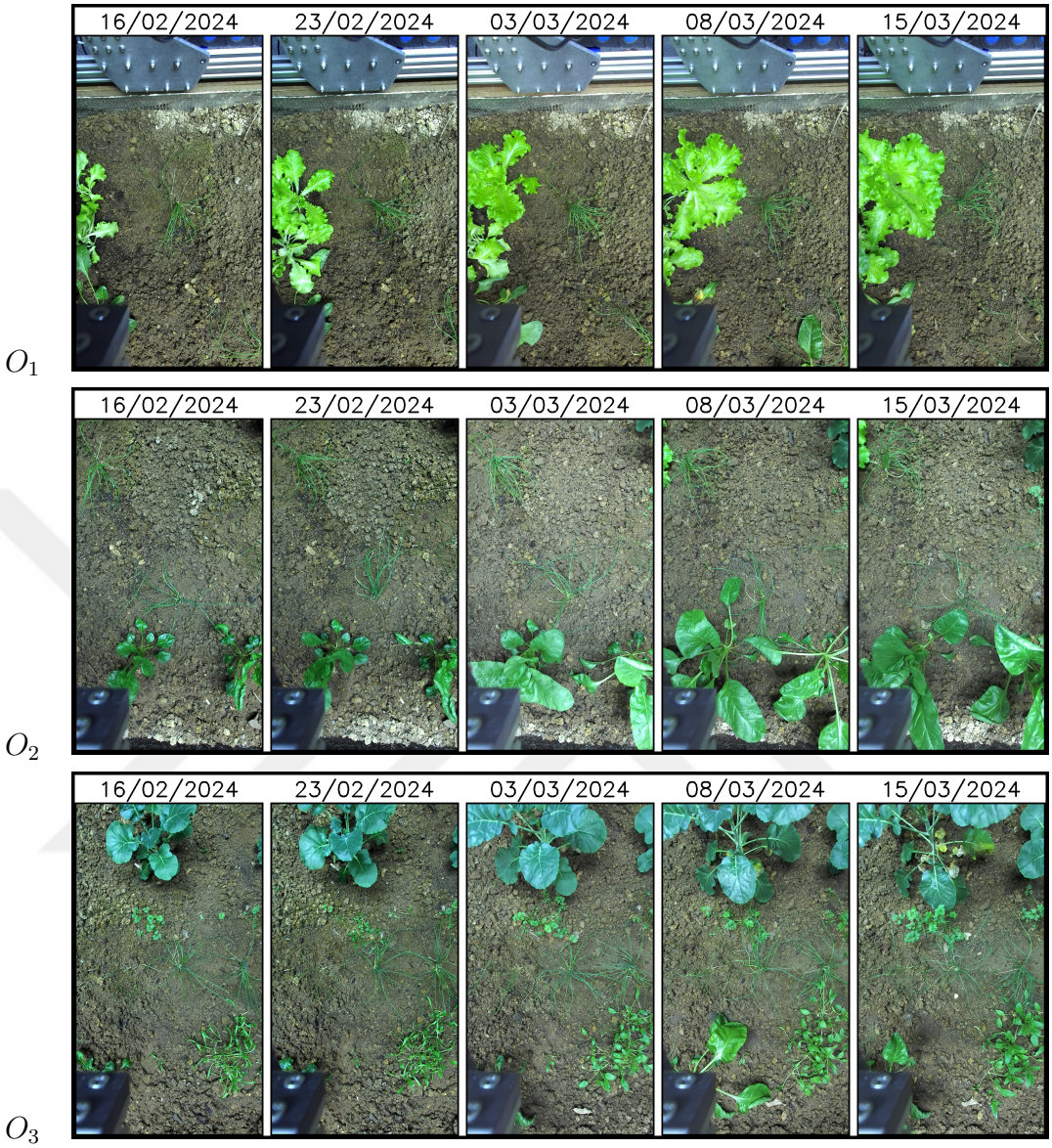


Figure 2.9. RGB data of spring onion plants vs time.

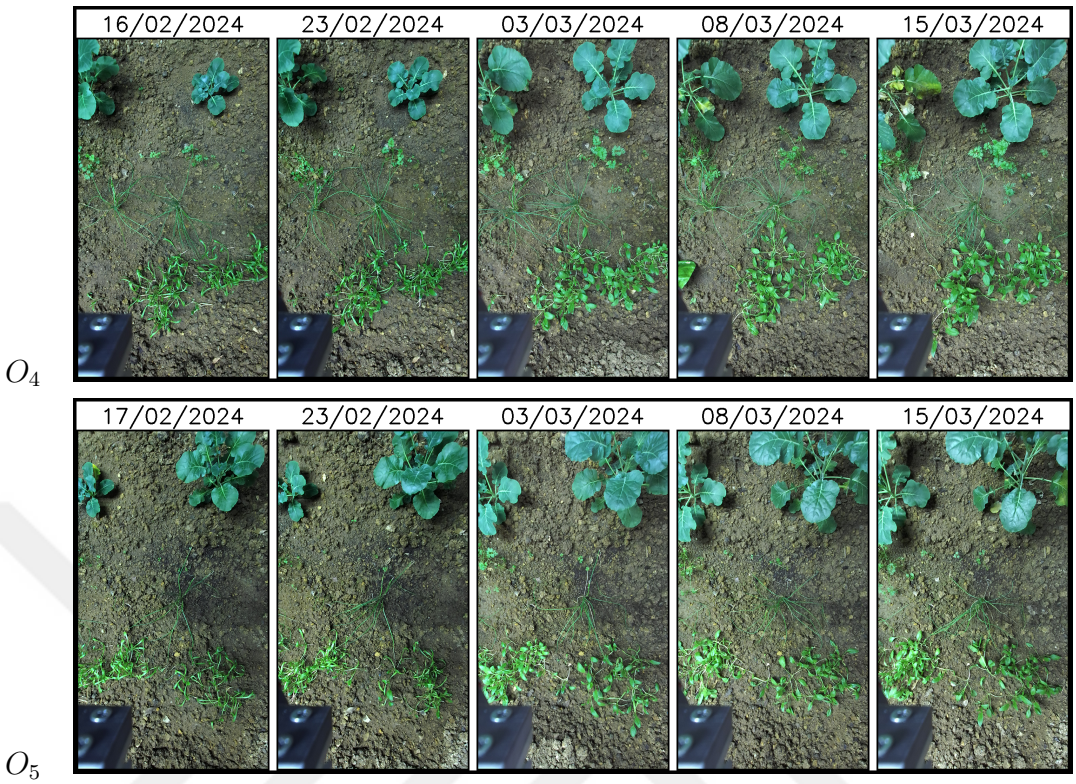


Figure 2.10. RGB data of spring onion plants vs time (cont.).

Spring onions are seen in Figure 2.9 and Figure 2.10. As observed, other plants have extended to their sown areas. Interestingly, spring onions did not grow. Sown O_1 and O_5 plants grew slightly, O_3 and O_4 remained constant in size while the number of leaves of O_2 decreased over time. As such, spring onions could not be harvested.

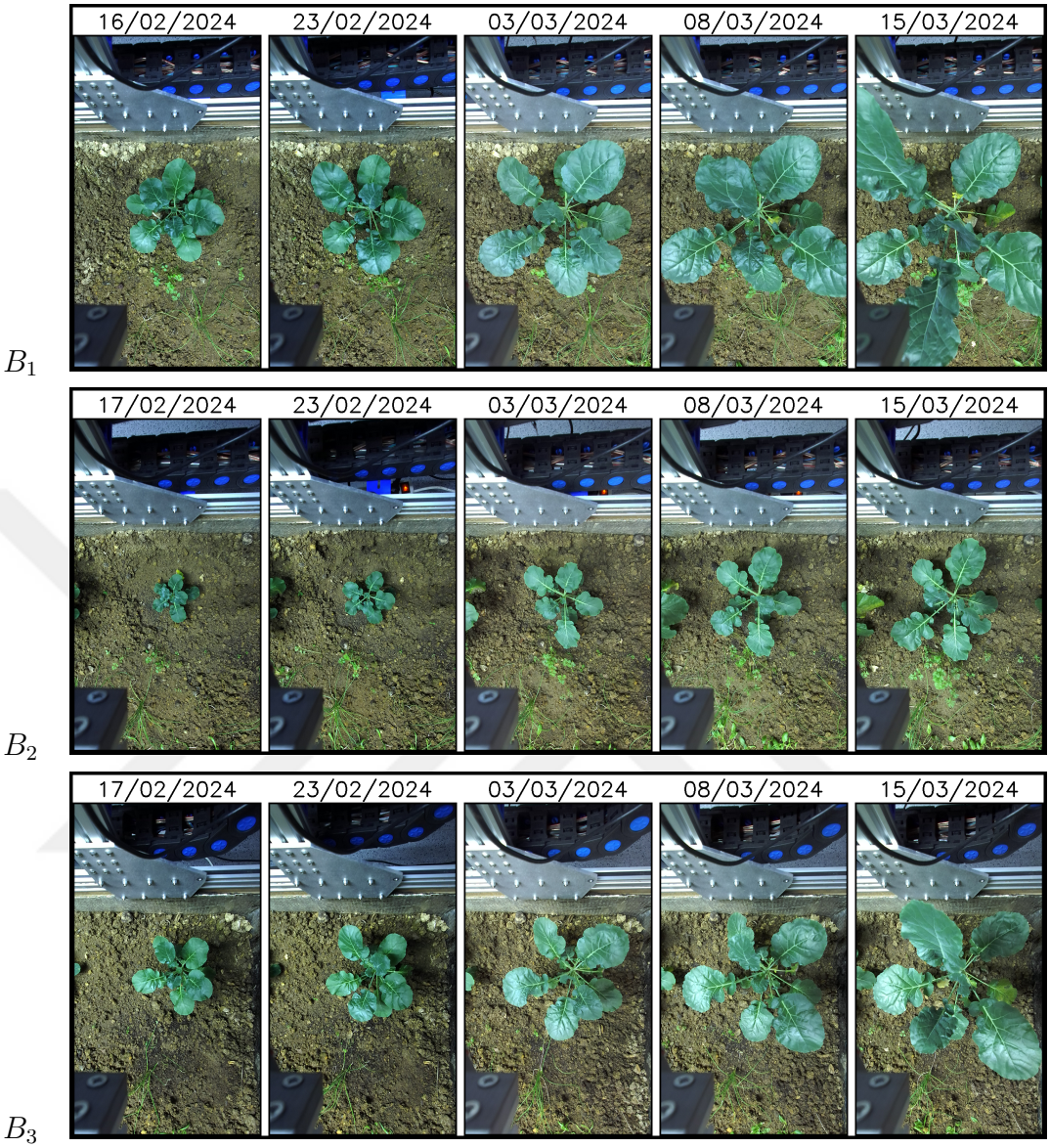


Figure 2.11. RGB data of broccoli plants vs time.

Figure 2.11 shows samples from the growth evolution of the planted broccolis. As observed, all grew healthily. As they were away from other plants, they were not subject to occlusion.

3. VISUAL PLANT DETECTION & SEGMENTATION

This chapter is focused on visual plant detection. There are three parts to this work:

- Coarse and fine segmentation of individual plants regardless of their species
- Detection of leaves on the segmented plants
- Learning dataset preparation

This chapter is organized as follows: First, related literature is reviewed briefly. Next, the training of the YOLO models and the preparation of the datasets used for this training are explained. In the last section, experimental results are evaluated.

3.1. Related Literature

Machine vision techniques have become increasingly popular for monitoring plants. One of the main reasons for this is that machine vision techniques can perform these tasks without causing physical damage to plants [9]. Some studies go beyond just using RGB images and use depth data obtained from stereo cameras. These studies estimate plant traits such as height, leaf area, leaf color, and orientation from point clouds or depth images alone [26–29].

Initially, researchers used traditional machine learning methods on RGB images of plants. Recently, deep learning methods have become more prominent - as their performance is expected to be better in plant detection and segmentation. CNNs, Mask R-CNN, and YOLO are popular alternatives that have been used. These have been trained to decide whether plants are ready for harvesting or to estimate traits like plant area, leaf area, and number of leaves. For example, convolutional neural networks (CNNs) have been shown to work well in computing plant features [12, 30]. The choice of model often depends on what's more important: accuracy or speed. Larger CNN

structures are used when accuracy is the top priority. YOLO is chosen when speed matters more. Each model typically works on only one plant species [5, 12, 31–33].

One of the challenges in using CNNs is due to the data requirements pertaining to training and the lack of such datasets associated with plant detection and segmentation [34]. Many researchers have had to create their own relatively small datasets. They often grow plants themselves and take a few hundred or thousand pictures. This limits the size and variety of the data available [5, 33, 35]. Furthermore, even if a dataset exists, image annotation is a time-consuming process that requires a lot of work. Moreover, segmentation annotation which involves outlining specific parts of a plant in an image requires much more human labor than detection annotation. Besides open-source and fully annotated datasets [36], semi-automated annotation methods utilizing large zero-shot transformer models have been developed [37–39]. Furthermore, the data transfer learning and fine-tuning pre-trained architectures for specific tasks often need to be employed - as the amount of data is not typically large enough for training from scratch [33, 34, 40].

3.2. RGB-based Plant Detection & Segmentation

YOLO models were used for learning RGB-based detection and segmentation. In particular, YOLOv8 model was used. As such, the resulting system can be relatively light and fast.

3.2.1. Learning Plant Detection and Segmentation Models

The goal is to have a model that can detect and segment all the plants in the polyculture garden regardless of their species. As this is not possible with the existing YOLOv8m-seg model, the YOLOv8 model pre-trained with the COCO dataset was fine-tuned with our plant detection dataset $\mathcal{D}_P$ (Table C.1 in Appendix C). This training was done with 300 epochs and 50 epochs of patience. The performance metrics for the detection bounding boxes are given in Table 3.1. YOLOv8 models use a fitness

formula to determine the best weights of the model as

$$\text{Fitness Score} = \sum_i W^T m_i \tag{3.1}$$

where $W = \begin{bmatrix} 0.0 & 0.0 & 0.1 & 0.9 \end{bmatrix}^T$ is the vector of weights and $m \in \mathbf{R}^4$ is the vector of the following computed performance metrics:

- Precision - P
- Recall - R
- Mean Average Precision (mAP) - mAP0.5
- Mean Average Precision (mAP) - mAP0.5 : 0.95

Note that mAP0.5 refers to the average precision at which the IoU between the predicted bounding box and the ground truth box exceeds 0.5. For each class, AP is calculated, and then the average of these AP values across all classes is taken to get the mAP. The value mAP0.5 : 0.95 refers to the average precision that is calculated for multiple IoU thresholds ranging from 0.5 to 0.95 (at increments of 0.05). For each IoU threshold, AP is computed for each class, and then the APs are averaged over all classes and all thresholds to obtain the mAP. A patience value of 50 epochs was used to prevent overfitting on the dataset while training the models. If there was no improvement in fitness score for 50 epochs, training was stopped.

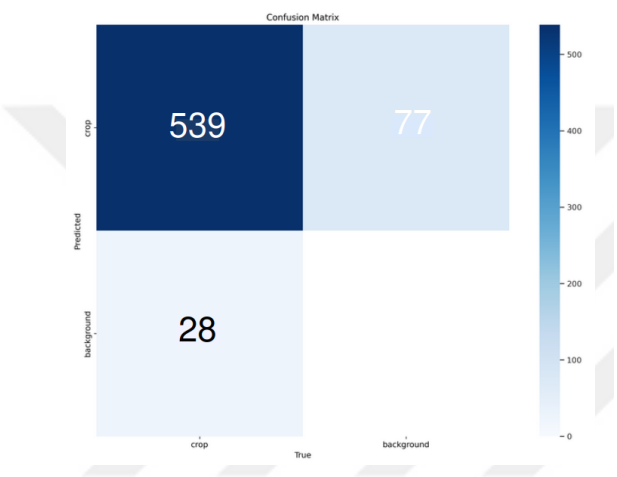
Table 3.1. Training performance of the plant detection.

Epoch	**Precision**	**Recall**	**mAP@50**	**mAP@50-95**
78	0.93	0.93	0.96	0.87

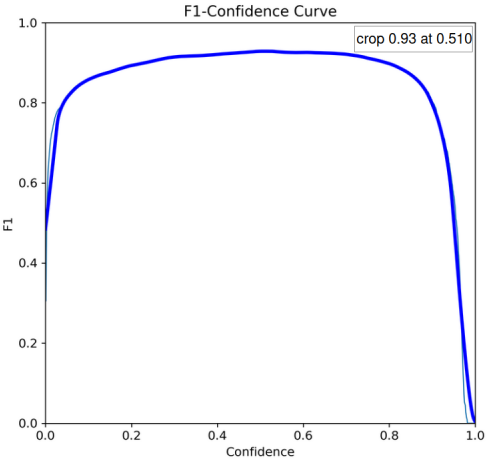
Since the YOLOv8 model works on both detection and segmentation tasks, it has separate performance metrics for the two tasks. The performance metrics for the segmentation masks are given in Table 3.2. The highest fitness score for the model was reached in the 78th epoch and training was stopped early in the 128th epoch.

Table 3.2. Training performance of the plant segmentation.

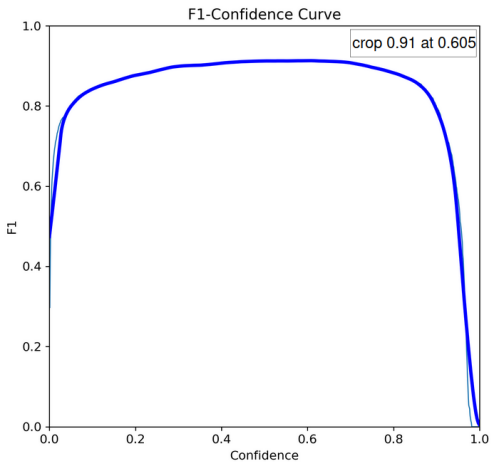
Epoch	Precision	Recall	mAP@50	mAP@50-95
78	0.93	0.90	0.94	0.77



(a) Confusion matrix of YOLOv8 plant detector.



(b) F1-Confidence curve for detection.



(c) F1-Confidence curve.

Figure 3.1. YOLOv8 plant detection-segmentation model training results.

The resulting confusion matrix, the F1-confidence curve for the detection task, and the F1-confidence curve for the segmentation mask are shown Figure 3.1. The precision and recall rates of the plant detection model are high. Additionally, as the

confusion matrix shows, the number of false positives is higher than false negatives. As the F1-Score chart shows, the model performs approximately the same over a wide range of confidence rates.

3.2.2. Plant Tracking

It should be recalled that the robot collects data of each plant by aligning its camera exactly above it. Furthermore, plants have been sown while adhering to the recommended seeding distances. As such, one would expect to have one plant per frame. However, this is rarely the case. Rather, the field of the RGB-D camera can possibly cover multiple plants as plants grow. Furthermore, plants exhibit nastic movement and turgor movement depending on water and movement toward light during the day. This means that a plant may move to the area of another plant. This means that there may be multiple detections and plant segments within each frame. Indeed, the YOLOv8 model will possibly detect and segment all the plants in an input RGB frame - if the confidence rate threshold is exceeded. However, the plant root stays in the same position and it is approximately located in the center of the bounding box. Hence, only the most centered plant detection can be kept and the remaining detections can simply be disregarded. This is achieved by considering the bounding boxes $\mathcal{B} = b_1, b_2, ..., b_N$ associated with the detected plants and finding the plant with the most centric bounding box b_{i^*}. Note that the center of each bounding box b_i is defined as

$$C_{b_i} = \left(\frac{x_1^i + x_2^i}{2}, \frac{y_1^i + y_2^i}{2} \right). \tag{3.2}$$

where (x_1^i, y_1^i) refers to the coordinates of its upper-left corner and (x_2^i, y_2^i) refers to the coordinates of lower-right corner. l The distance d_i between each box center and the image center is computed using the Euclidean norm as

$$d_i = |C_{b_i} - C_I| = \sqrt{\left(\frac{x_1^i + x_2^i}{2} - c_x \right)^2 + \left(\frac{y_1^i + y_2^i}{2} - c_y \right)^2}. \tag{3.3}$$

The index of the most centered bounding box i^* is determined by

$$i^* = \arg\min_{i \in 1,\ldots,N} d_i. \tag{3.4}$$



(a) Resulting image after the YOLOv8 model's prediction.



(b) Resulting image after applying the plant track algorithm on YOLO predictions.

Figure 3.2. Plant track algorithm result.

A sample case is as shown in Figure 3.2(a). Note that the trained YOLOv8 model finds multiple plant detections and segmentations. Only the tracked plant is shown in Figure 3.2(b).

3.3. RGB-based Leaf Detection

Leaf detection is done considering plant detection and segmentation results. For this, YOLOv7 and YOLOv8 models were trained and their performances were compared. The inputs of both are RGB data corresponding to each detected plant as obtained by removing the remaining background from the plant's segmentation mask.

Three YOLO models were trained - each one for 300 epochs. As the YOLOv8 model has a native patience mechanism for early stopping, there are 50 epochs associated with this.

(i) Fine-tuned YOLOv7: YOLOv7 model was fine-tuned model with the manually annotated leaf detection dataset $\mathcal{D}_L$ as given in Table C.2 in Appendix C.

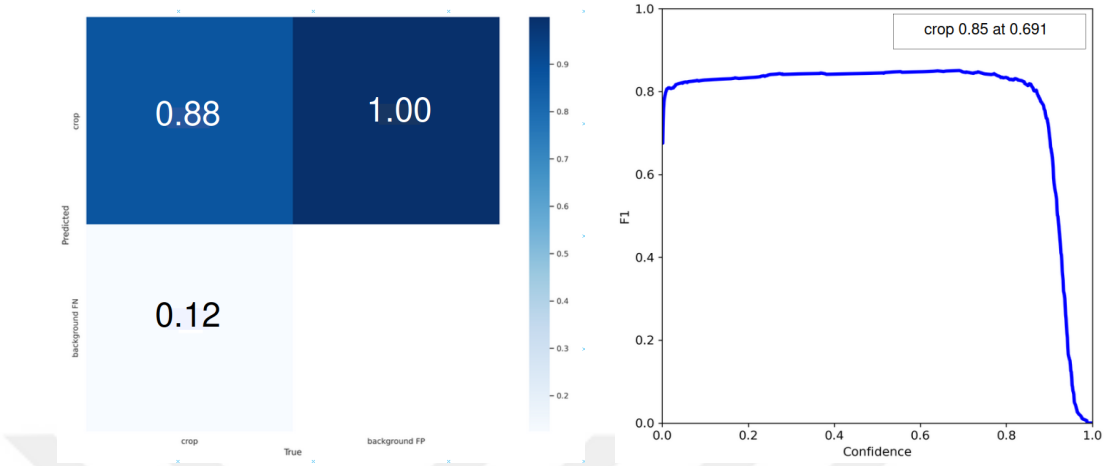
(ii) Trained YOLOv8m: YOLOv8m is trained further using the external leaf detection dataset $\mathcal{D}_E$ (Table C.3 in Appendix C). Note that the initial weights of YOLOv8m have been pretrained with the COCO dataset.

(iii) Fine-tuned YOLOv8m: YOLOv8m is first trained with external leaf detection dataset $\mathcal{D}_E$ followed by fine-tuning using the manually annotated leaf detection dataset $\mathcal{D}_L$. Again, the initial weights of YOLOv8m have been pretrained with the COCO dataset.

After the training was completed, performance was assessed and the best model was chosen. The performance metrics of fine-tuned YOLOv7 model training are given in Table 3.3. Since there is no early stoppage mechanism, training continued for 300 epochs. The highest fitness score for the model was reached at the 61th epoch.

Table 3.3. Fine-tuned YOLOv7 - Leaf detection training performance.

Epoch	**Precision**	**Recall**	**mAP@50**	**mAP@50-95**
61	0.87	0.85	0.88	0.49



(a) Confusion matrix of the YOLOv7 leaf detector.

(b) F1-Confidence curve for detection.

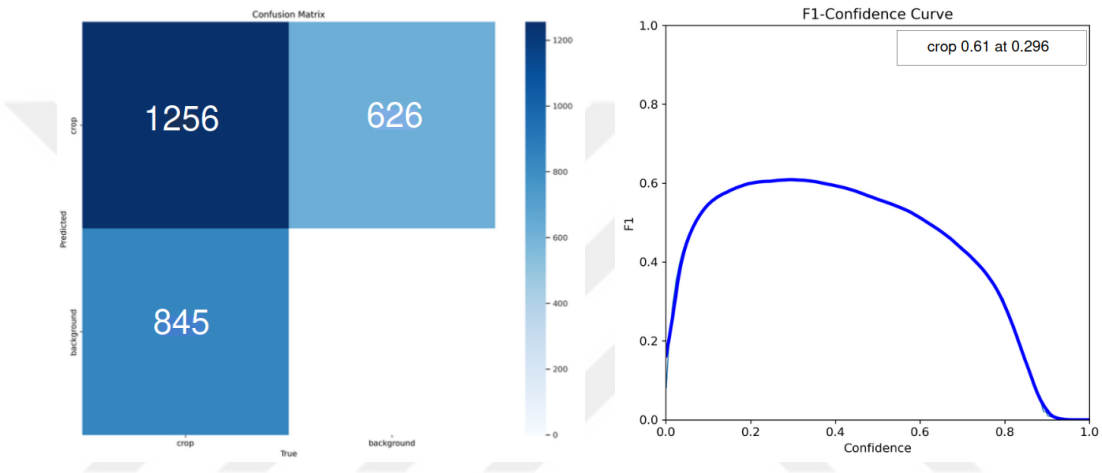
Figure 3.3. YOLOv7 leaf detection model training results.

The confusion matrix and F1-Confidence curve from leaf, detection training is seen in Figure 3.3. After retraining, the pretrained model achieved improved precision and recall values. Figure 3.3(a) demonstrates the confusion matrix of the model's training result and Figure 3.3(b) demonstrates the confusion matrix of the model's training result. Accurate leaf detection has proven to be a more challenging task than plant detection - even if it is performed only on individually masked plants. This is attributed to the large variation in leaf shapes and the tendency of leaves to obscure one another.

The performance metrics of the trained YOLOv8 leaf detector training are given in Table 3.4. As this model has an early stopping feature, this is set as 50 epochs. The highest fitness score for the model was attained at the 33rd epoch. Its training stopped at the 83rd epoch after 50 epochs without improvement.

Table 3.4. Trained YOLOv8 - Leaf detection training performance.

Epoch	Precision	Recall	mAP@50	mAP@50-95
33	0.67	0.56	0.61	0.35



(a) Confusion matrix of YOLOv8. (b) F1-Confidence curve.

Figure 3.4. YOLOv8 leaf detection model training results.

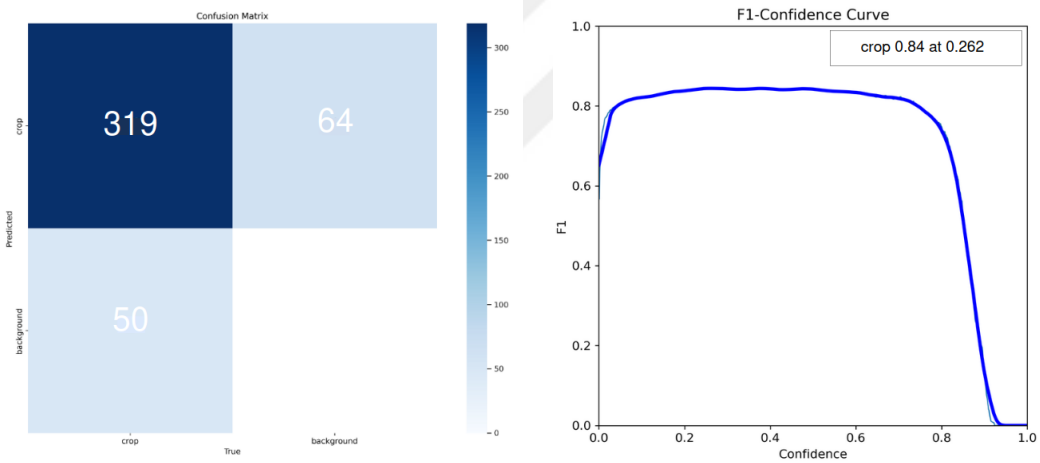
The confusion matrix and the F1-Confidence curve associated with the leaf detection training are shown in Figure 3.4. Figure 3.4(a) demonstrates the confusion matrix of the model's training result and Figure 3.4(b) demonstrates the confusion matrix of the model's training result. The trained model has low precision and recall values. The most important reason for this is that the training dataset contains many different types. The model could not reach high precision and recall values, but it gained intuition on plant leaves. Instead of using this model directly, fine-tuning it on the TarRob dataset is necessary to accurately estimate the number of leaves of plants in the polyculture garden.

The performance metrics of the fine-tuned YOLOv8 leaf detector model are given in Table 3.5. Again, this model stops training if there are no improvements in 50 epochs. The best performance was saved at the 76th epoch. The highest fitness score

for the model was reached at the 33rd epoch. The training stopped at the 125th epoch after 50 epochs without improvement. Figure 3.5 shows the confusion matrix as a result of training, the F1-Confidence curve for the leaf detection task. Figure 3.5(a) demonstrates the confusion matrix of the model's training result and Figure 3.5(b) demonstrates the confusion matrix of the model's training result.

Table 3.5. Fine-tuned YOLOv8 - Leaf detection training performance.

Epoch	**Precision**	**Recall**	**mAP@50**	**mAP@50-95**
76	0.85	0.84	0.88	0.54



(a) Confusion matrix. (b) F1-Confidence curve for detection.

Figure 3.5. Fine-tuned YOLOv8 leaf detection model training results.

It is observed that the YOLOv8 model trained with an external dataset has the lowest performance metrics among the three models. When the confidence rate is increased, the recall value decreases rapidly. It performs best at a confidence rate of 0.298. The external dataset contains many plant species collected from different environments and therefore the model is not complex enough to achieve high precision and recall values.

In fine-tuned models, the YOLOv7 model performed slightly better in precision and recall performance than the YOLOv8. The YOLOv8 model is slightly more successful in mAP@50-95 results. Figure 3.6(a) shows the detections of the YOLOv7 model on a sample broccoli image. Figure 3.6(b) shows the detections of the YOLOv8 model on the same image. Among the systems showing similar performance metrics, the YOLOv8 model was preferred. Although the YOLOv7 model predicts leaves at least as accurately as the YOLOv8 model, Figure 3.6 shows that during training, the anchor boxes of the YOLOv8 model were better refined, and the model generates more accurate bounding boxes around the leaves.



(a) Fine-tuned YOLOv7 (b) Fine-tuned YOLOv8

Figure 3.6. Leaf detection examples of the trained models.

3.4. TarRob Plant Detection & Segmentation Dataset

The training of plant detection and segmentation models both require learning datasets that can be used in polyculture garden settings. As there are no such datasets, these were prepared within the course of this thesis. Their preparation consisted of the

following stages:

- Dataset annotation
- Dataset augmentation
- Dataset cleaning

3.4.1. Dataset Annotation

The dataset annotation is associated with two tasks:

- The annotation of bounding boxes
- The annotation of segmentation masks.

For the first task, there were three alternatives: i) Manual labeling, ii) Detection with a pre-trained model, and iii) Zero-shot detection. Instead of classification of the plant or leaf species, the goal is to have just one class for both detectors as crop and leaf. First, we consider detection with a zero-shot detection. For this, Roboflow has built-in Grounding DINO - SAM pipeline [41]. It is observed that Grounding DINO detects several false positives in a low confidence threshold scenario while recall value is low either for one class or for classes of each plant species in a high confidence threshold scenario. Therefore Grounding the DINO model is not suitable to detect plants. Hence, either manual labeling or pre-trained detector needs to be considered. If the detector works well enough or the dataset is not too complex, using a pre-trained detector is the better option. However, if the dataset is complex, manual labeling is needed.

For the second task, manual labeling is not an option due to its timewise high cost. Rather, an automated approach needs to be utilized. For this, the SAM model seemed viable [38]. This is a visual transformer model that accepts bounding boxes as prompts. SAM model performed well enough to be our automated segmentation annotation tool.

Figure 3.7 shows the resulting data annotation pipeline. The first model detects plants and outputs bounding boxes. A script was created for both manual labeling and pre-trained detectors. The script starts the pipeline in both cases and creates annotations. The resulting bounding boxes are then input as prompts to the following SAM model. SAM model finds segmentation masks of the plants and masks become the inputs for the second detector model. In the last step, the pipeline gives segmentation masks of the leaves as output. Figure 3.8 shows the result of the full pipeline on an example broccoli image.

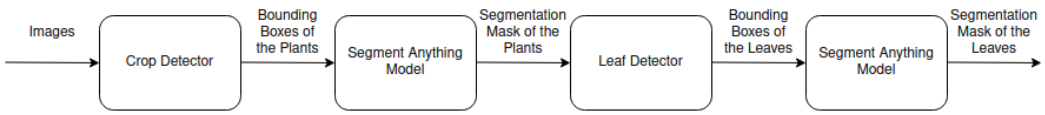
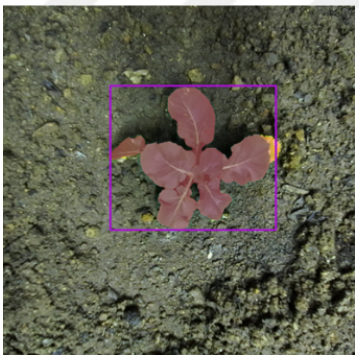


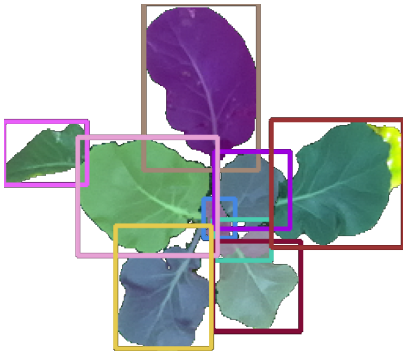
Figure 3.7. Data annotation pipeline.



(a) Detected and segmented plant.



(b) Segmentation mask in the bounding box.



(c) Detected and segmented leaves.

Figure 3.8. Data annotation pipeline on an example broccoli image.

Figure 3.8(a) shows the result of plant segmentation with using SAM model after detection. Figure 3.8(b) shows the result of the second step of the annotation pipeline. After finding segmentation mask of the plant background was removed. Figure 3.8(c) shows the leaf detection and segmentation results of the pipeline. For segmentation task, SAM model was used. However, for the leaf detection task another detector which was trained to detect plant leaves was used.

3.4.2. Data Augmentation

The dataset is augmented by applying a random selection of augmentations to each image in the dataset. A sample case is illustrated in Figure 3.9. Figure 3.9 shows the original image and then two augmented versions of it. For each, three augmentation methods were chosen and applied. All augmentation methods have their own value intervals. The detailed information on augmentation for each dataset can be found in Appendix C. For instance, if rotation, brightness, and flip augmentations are chosen with a 2x multiple, each image will have a random selection of those three augmentations. The values are selected from their intervals. For example, 5% brightness, 10% clockwise rotation, and horizontal flip [42].



Figure 3.9. Augmentation example.

3.4.3. Dataset Cleaning

The dataset was cleaned by removing problematic data due to the following reasons:

- Problems in robot's sensing
- Localization problems

- Problems in lighting

Problems in the robot's sensing manifested as follows:

- Plants' sizes exceeded the camera's field of view
- Plants' heights became too tall for reliable depth sensing

In particular, the data collected between 09.02.2024 and 17.05.2024 did not have these problems. This filtration ensured only measurements with complete plant visibility and reliable depth data were used for phenotype computation. For measurement purposes, broccoli is the most suitable plant species. For chards and lettuces, stereo camera versions offer better plant phenotype computation.

This problem is also associated with camera issues such as the camera shutting down. This issue causes fully black images. Camera issues are as follows:

- From 15/04/2024 at 06:00 to 16/04/2024 at 18:00
- From 01/05/2024 at 20:00 to 03/05/2024 at 08:00
- On 05/05/2024 at 20:00

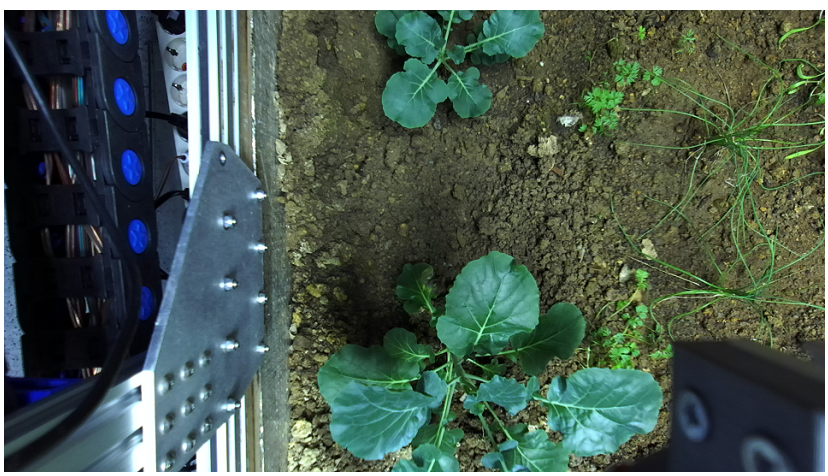


Figure 3.10. Sample RGB data with localization problem.

Localization problems are due to problems in the robot's movements when collecting the RGB-D data. Consequently, the robot's position may fail to align with the target plant as intended. Figure 3.10 shows such a case. The broccoli plant should be positioned at the center of the image, but it is currently located in the upper part. Due to this positioning issue, the plant cannot be seen entirely. Some parts of the broccoli are outside the field of view.

The following data are observed to suffer from this issue:

- From 13/02/2024 at 00:00 to 16/02/2024 at 18:00
- On 16/02/2024 at 20:00
- From 02/03/2024 at 00:00 to 02/03/2024 at 06:00
- On 03/03/2024 at 19:00

Additional localization problems affected plants farthest from the robot's home position, specifically 3 chard and 3 broccoli plants, excluding the first two rows of plants:

- From 19/02/2024 at 00:00 to 21/02/2024 at 12:00
- On 23/02/2024 at 00:00
- From 27/02/2024 at 00:00 to 02/03/2024 at 08:00



Figure 3.11. Sample RGB image with a lighting problem.

Lighting problems are attributed to the failure to turn on the LED lighting system due to loss of connection with the processor and in some cases, and in other cases by power outages in the building. Figure 3.11 shows a sample with a lighting problem. Images taken every six hours require us to take images at night as well. When there is a light problem, images cannot be taken of the plant at night. Lighting problems are noted to occur in the following data:

- On 21/02/2024 at 18:00

3.5. Experimental Results

This section presents the experimental results of plant detection and segmentation by the TarRob robot in the polyculture garden.

3.5.1. Plant Detection & Segmentation

The evolution of plant detection and segmentation for lettuce, chard, and broccoli plants is tracked over five weeks. The trained VOLOv8 model is highly accurate even in relatively challenging scenarios. As expected, the reliability of detection decreases as plants grow and enter neighboring plants' areas. The system has almost always been successful on lettuce, chard, and broccoli plants. The model showed lower performance after the March 15 images due to the camera position no longer being able to view the entire plant.

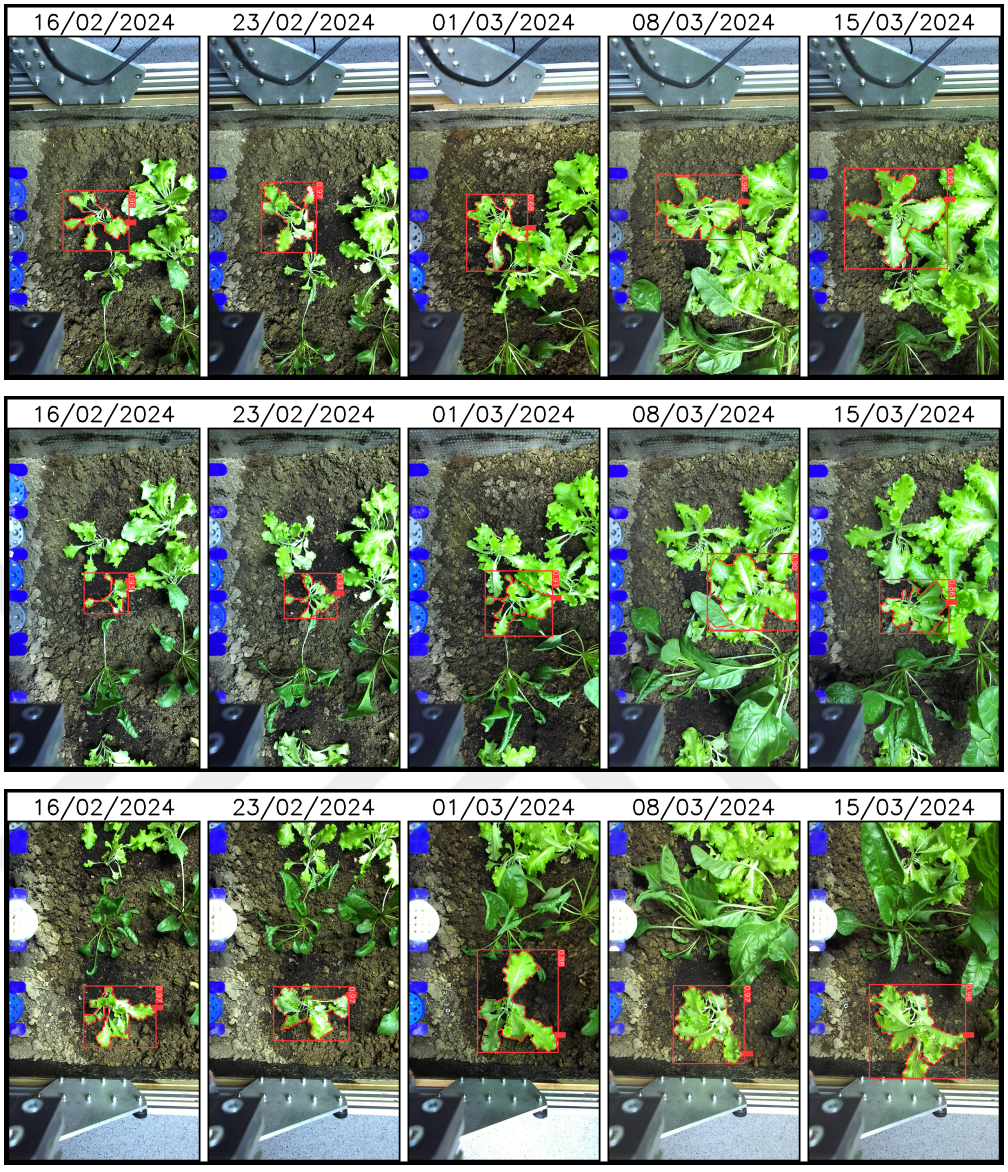


Figure 3.12. Detection and segmentation of lettuces vs time.

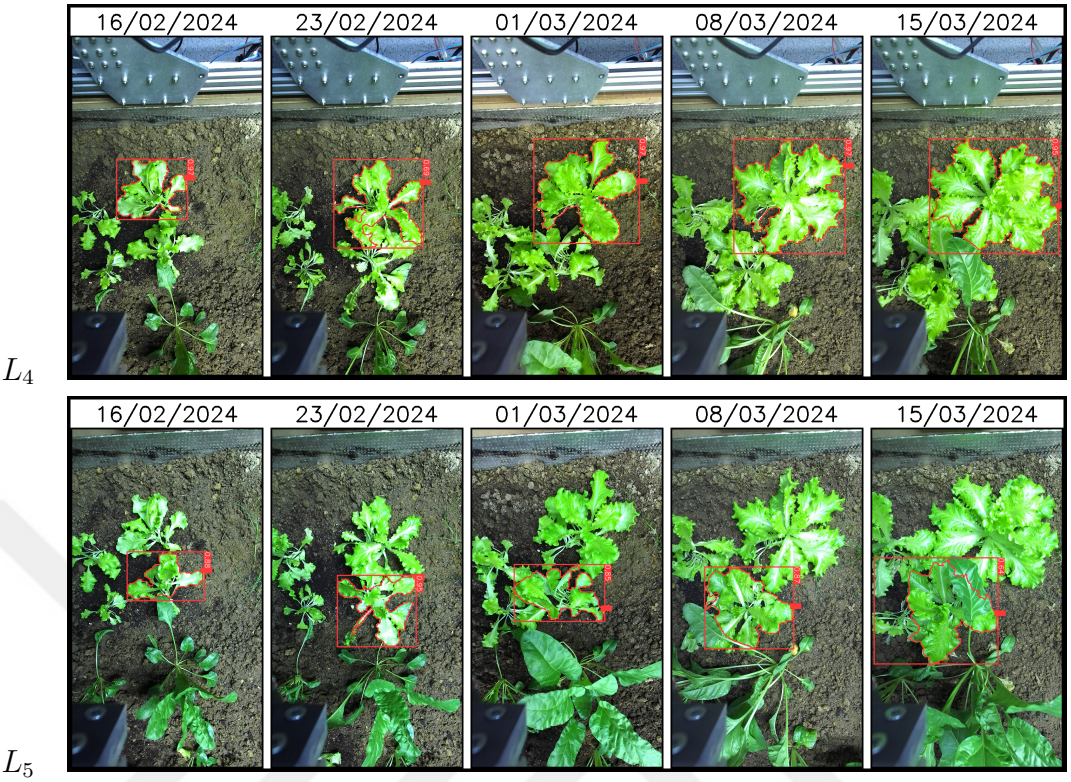


Figure 3.13. Detection and segmentation of lettuces vs time (cont.).

Figure 3.12, and Figure 3.13 show that detection and segmentation results are often quite accurate, even for lettuces that are intertwined. The YOLOv8 model accurately detected and segmented L_1 and L_3 in all weeks. L_2 was not detected by the model in the fourth week because it was covered by both lettuce plant L_5 and chard plant C_2. In the fifth week, some of them remained below L_5.

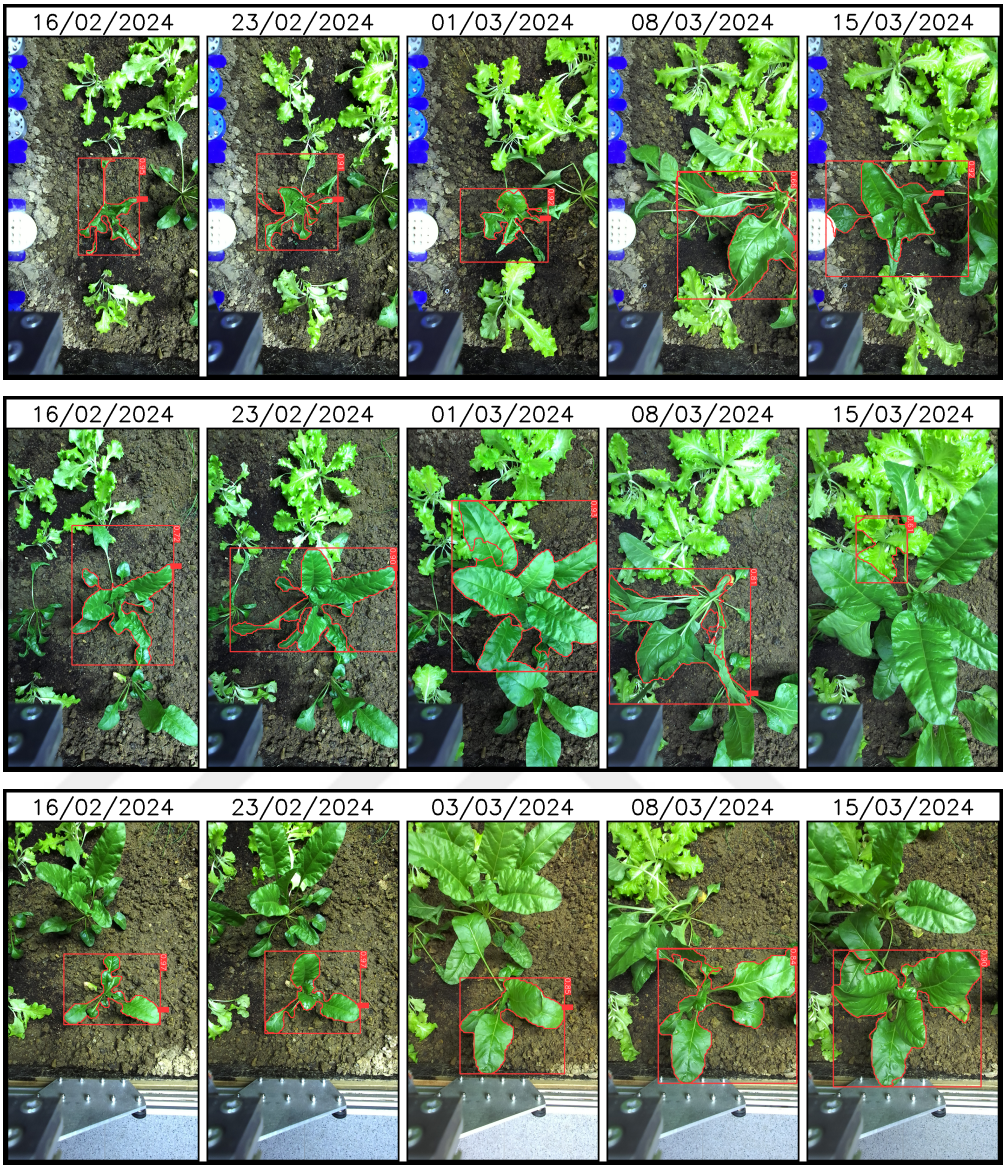


Figure 3.14. Detection and segmentation of chards vs time.

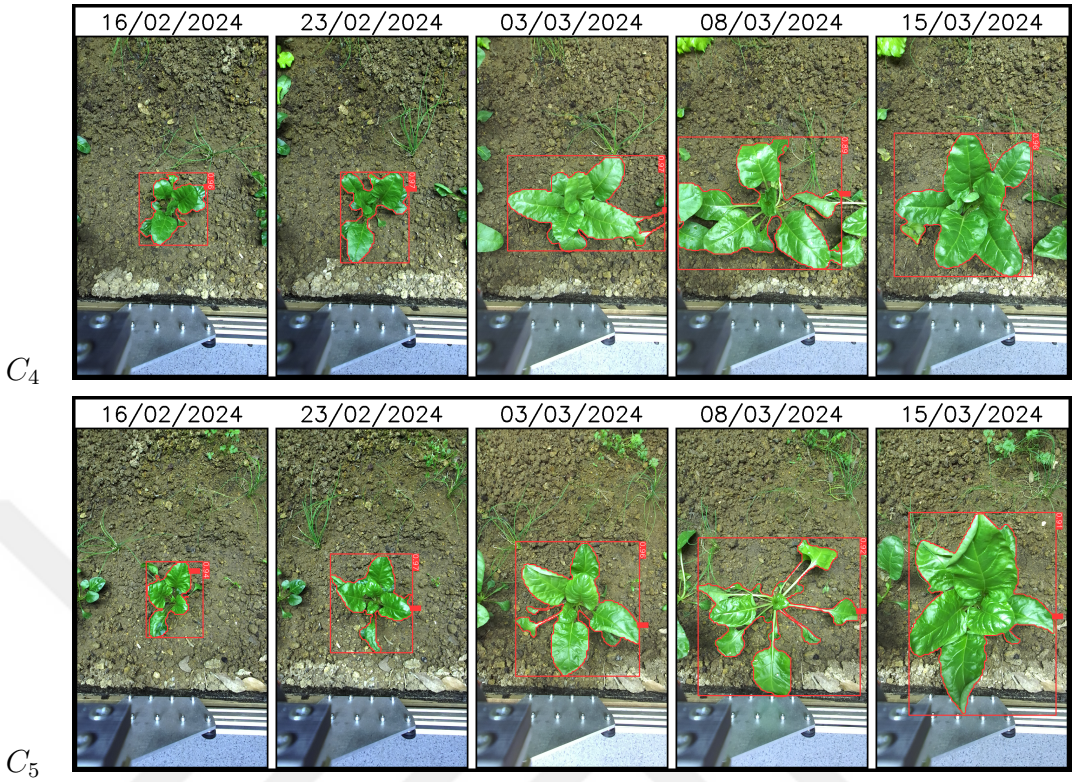


Figure 3.15. Detection and segmentation of chards vs time (cont.).

Figure 3.14, and Figure 3.15 show the detection and segmentation results of the model on chard images. The chard plant C_2 dominated the field, growing faster than other chards. In last week's image, it was not detected because it was too close to the camera. It also caused chard plant C_1 to go undetected in the last two weeks. In the fourth-week image of C_4, the leaves of other chards are wrongly taken as the leaves of chard plant C_4. Apart from these observations, the model performed accurate detection and segmentation on chard images.

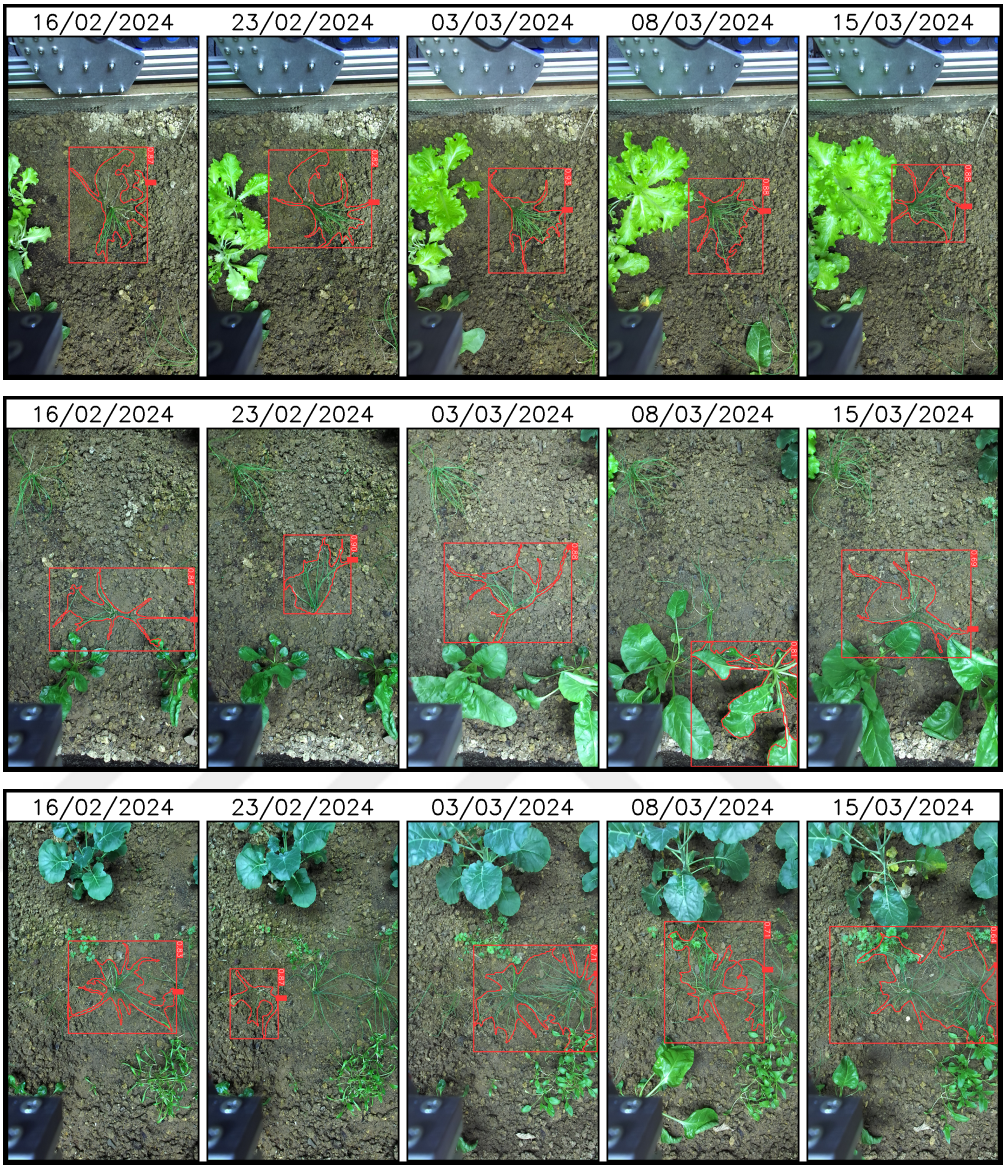


Figure 3.16. Detection and segmentation of spring onions vs time.

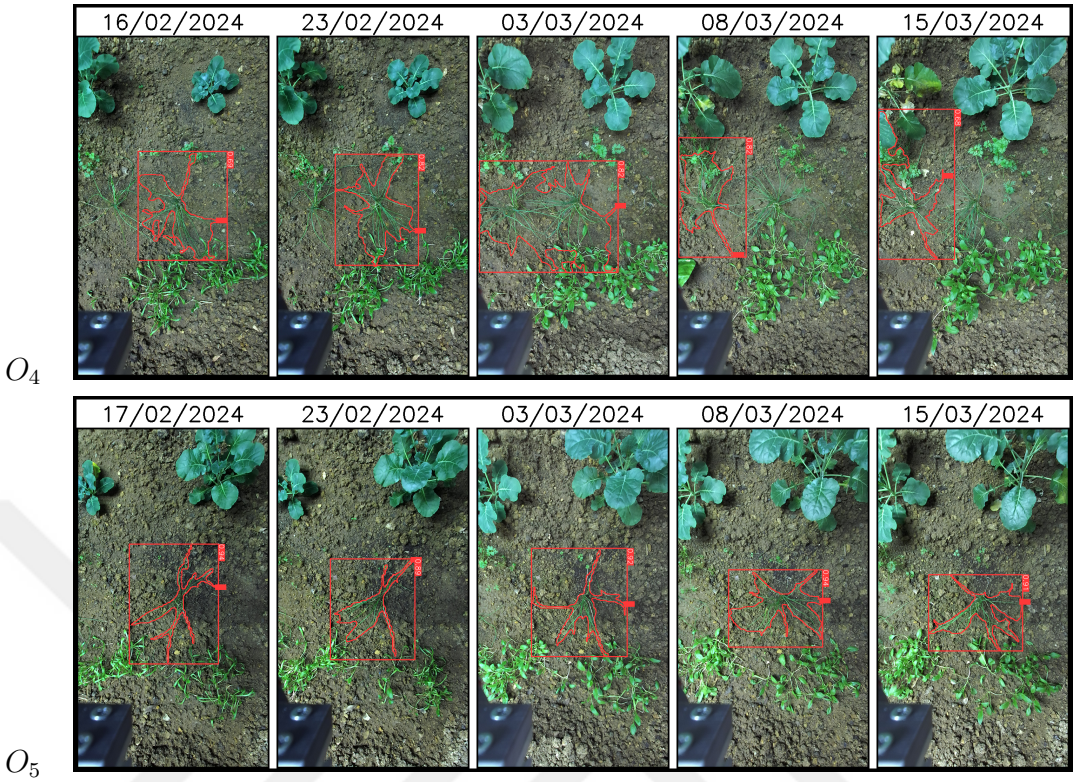


Figure 3.17. Detection and segmentation of spring onions vs time (cont.).

Figure 3.16, and Figure 3.17 show the detection and segmentation results of the model on spring onion images. Spring onion is the most difficult plant species to detect and has the lowest recall value. Although the model is accurate in detection tasks, the mAP values of the segmentation results are low. Most of the false negative results are observed to be those of spring onion. As the confidence rate increases, spring onions become harder to detect.

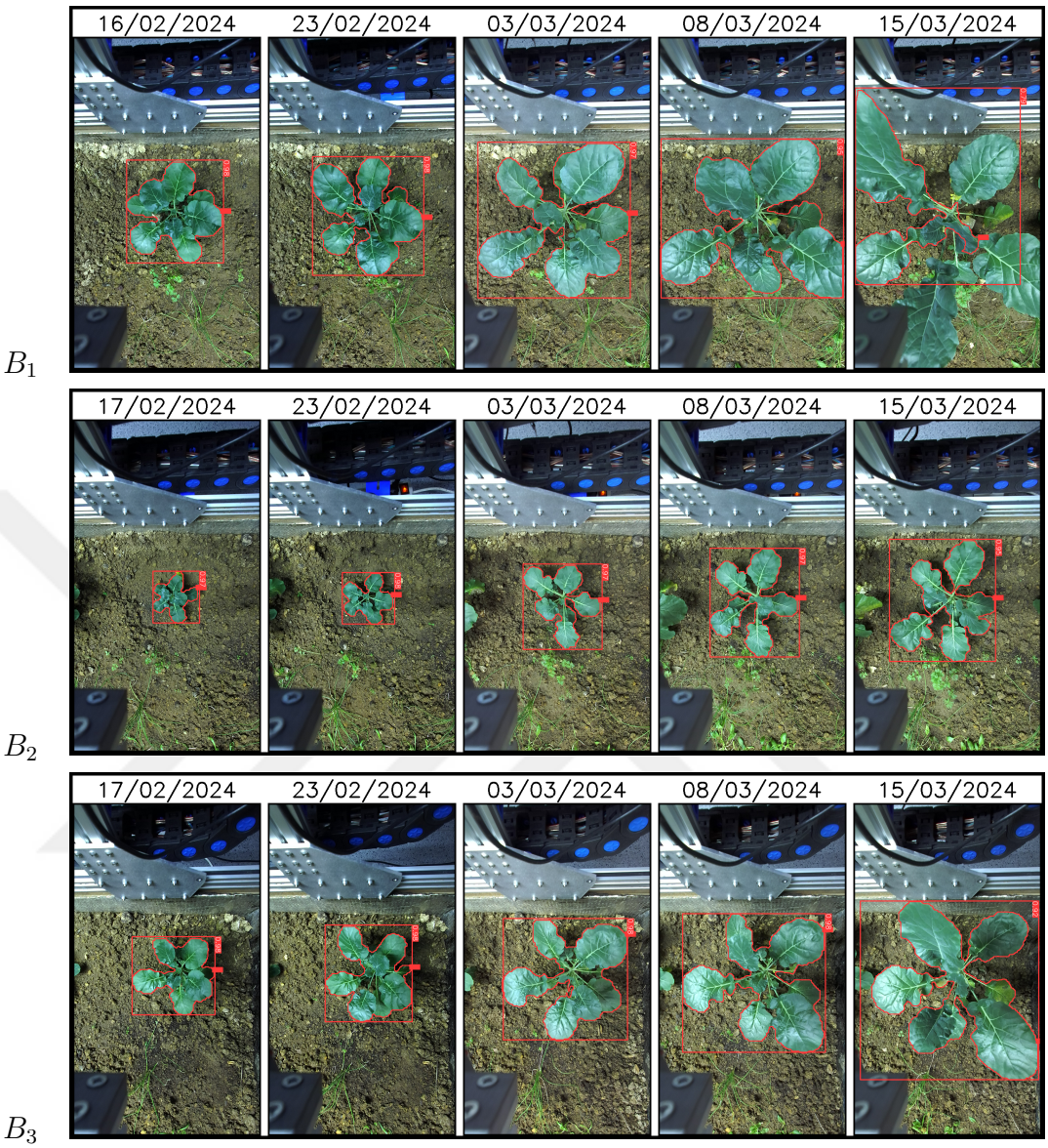


Figure 3.18. Detection and segmentation of broccolis vs time.

Figure 3.18 shows the detection and segmentation results of the model on broccoli. Broccoli was detected and segmented perfectly accurately - except last week's B_1. The model could not detect and segment the broccoli plant B_1 as it had grown so much that it almost touched the camera and hence did not fit into the camera's field of view.

3.5.2. Leaf Detection

The leaf detection results of the YOLOv8 model are demonstrated in this subsection. It should be remarked that the number of canopy leaves is a highly changing

plant feature. Their visibility from observing above as done by the robot is affected by the flexible movements of the plant, their orientation towards the light, and the level of turgor pressure during data acquisition.

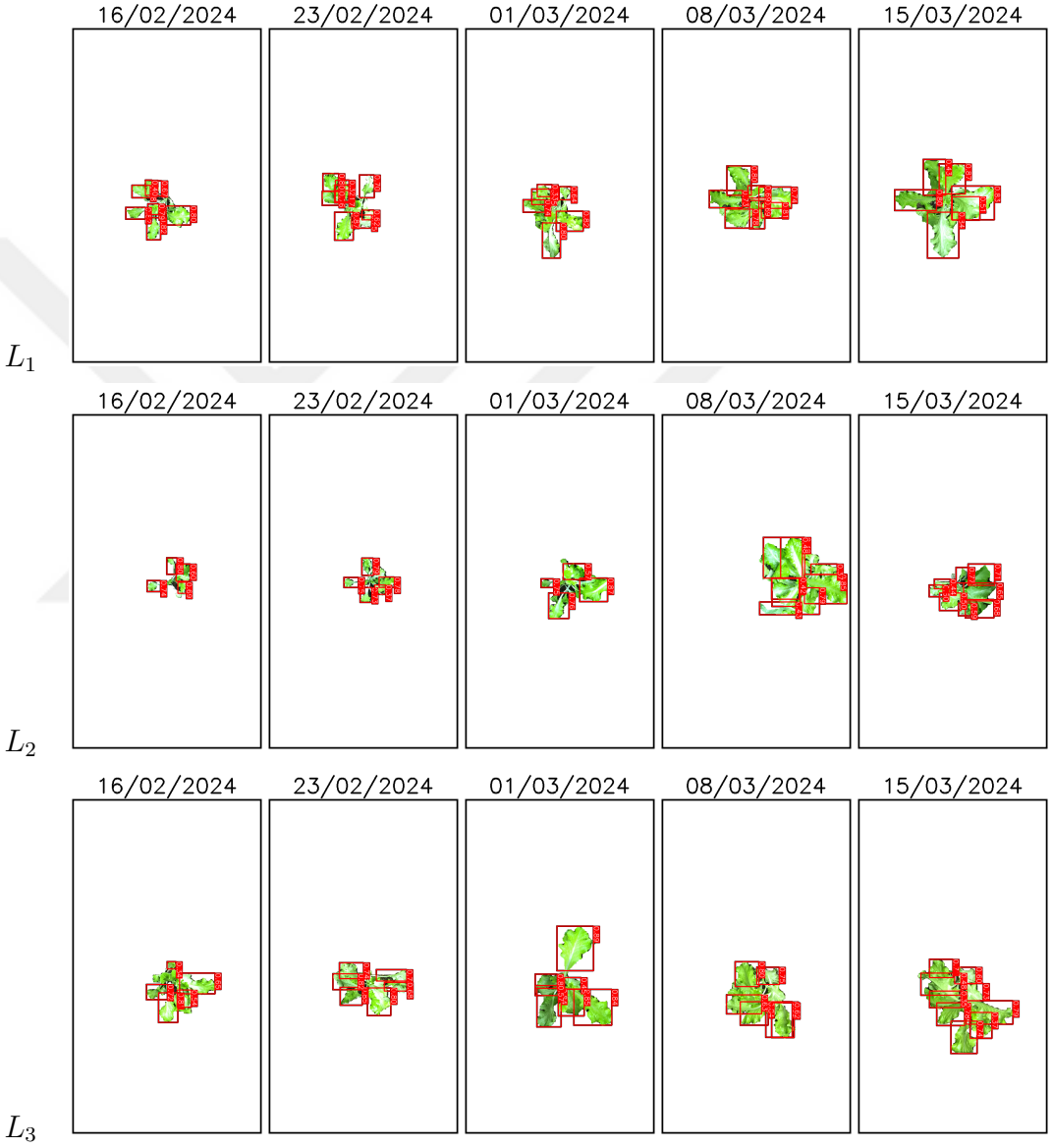


Figure 3.19. Detection of lettuce leaves vs time.



Figure 3.20. Detection of lettuce leaves vs time (cont.).

Figure 3.19 and Figure 3.20 suggest that the leaf detection model did not perform accurate detections on lettuce images. As the lettuce grows, it becomes easier to detect the leaves outside, but the fact that the plant has no stem and its leaves are intertwined makes leaf detection difficult.

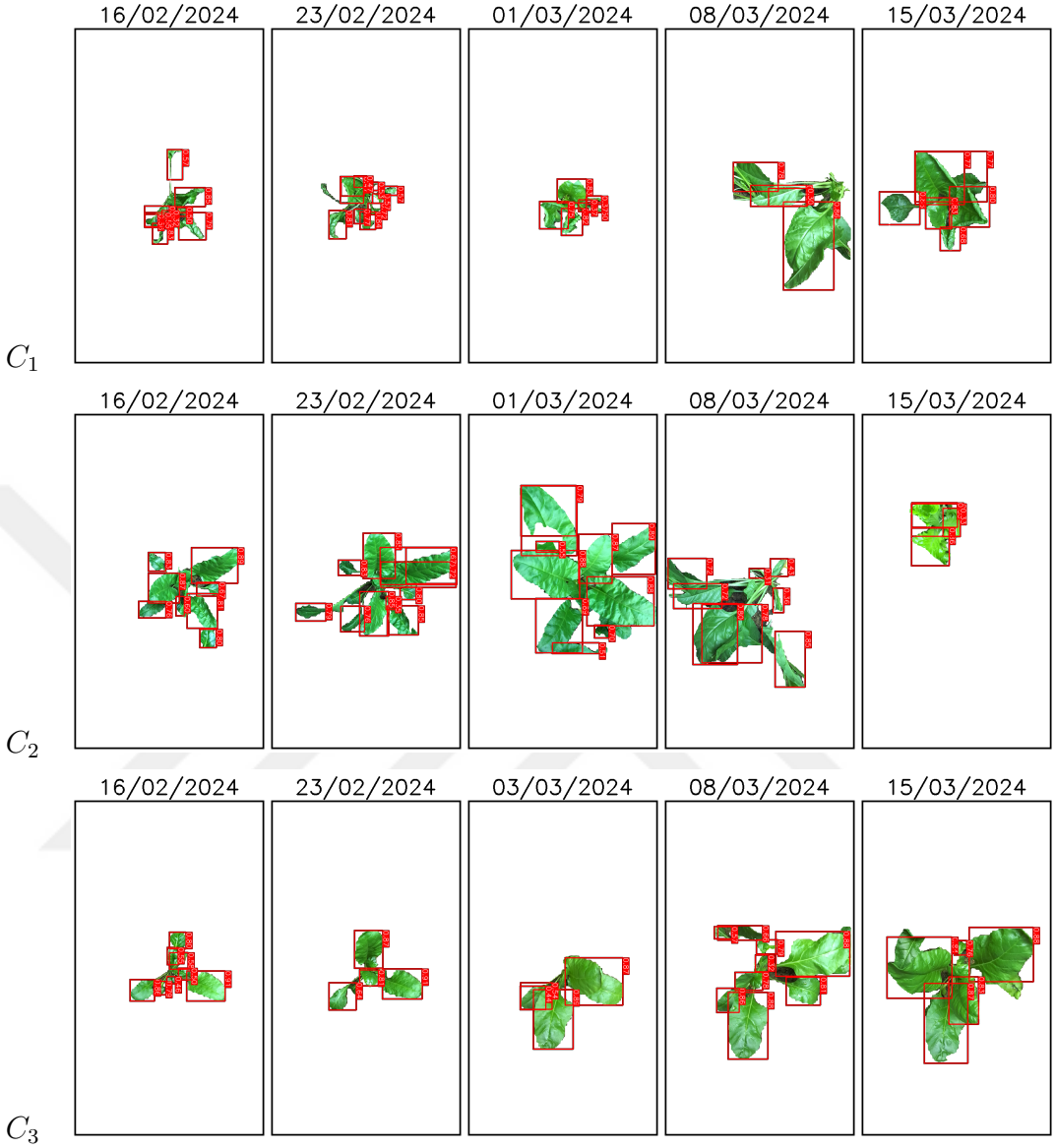


Figure 3.21. Detection of chard leaves vs time.



Figure 3.22. Detection of chard leaves vs time (cont.).

Figure 3.21 and Figure 3.22 reveal the leaf detection to perform reliably on chards - especially from the third week onwards. When the plant is in its vegetative stage, its height increases and its leaves grow upright. Therefore, the opened leaves cannot be seen in the image and are difficult to detect. Of course, if the plant is not correctly segmented, leaf detection will not be reliable. Indeed this is the case for the chard plant C_2 in the last week.



Figure 3.23. Detection of spring onion leaves vs time.

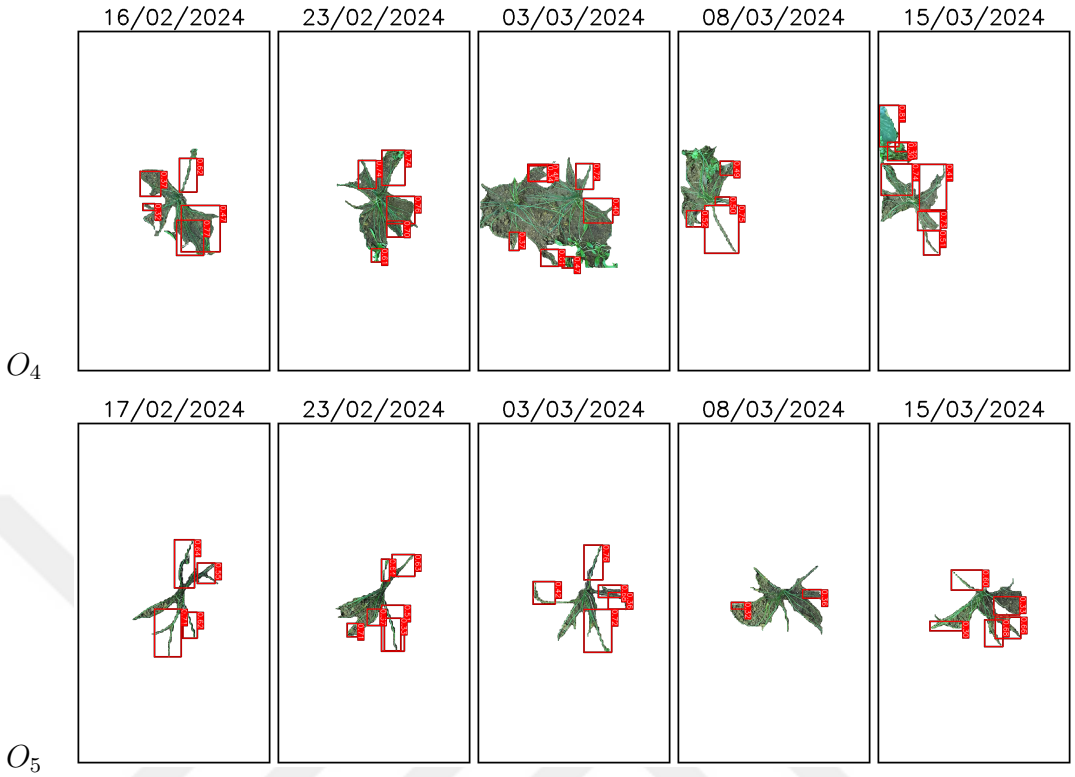


Figure 3.24. Detection of spring onion leaves vs time (cont.).

Figure 3.23 and Figure 3.24 show that leaf detection is an impossible task for spring onion due to its morphological structure. The lowest performance in all plant detection, segmentation, and leaf detection tasks was obtained on spring onion images.

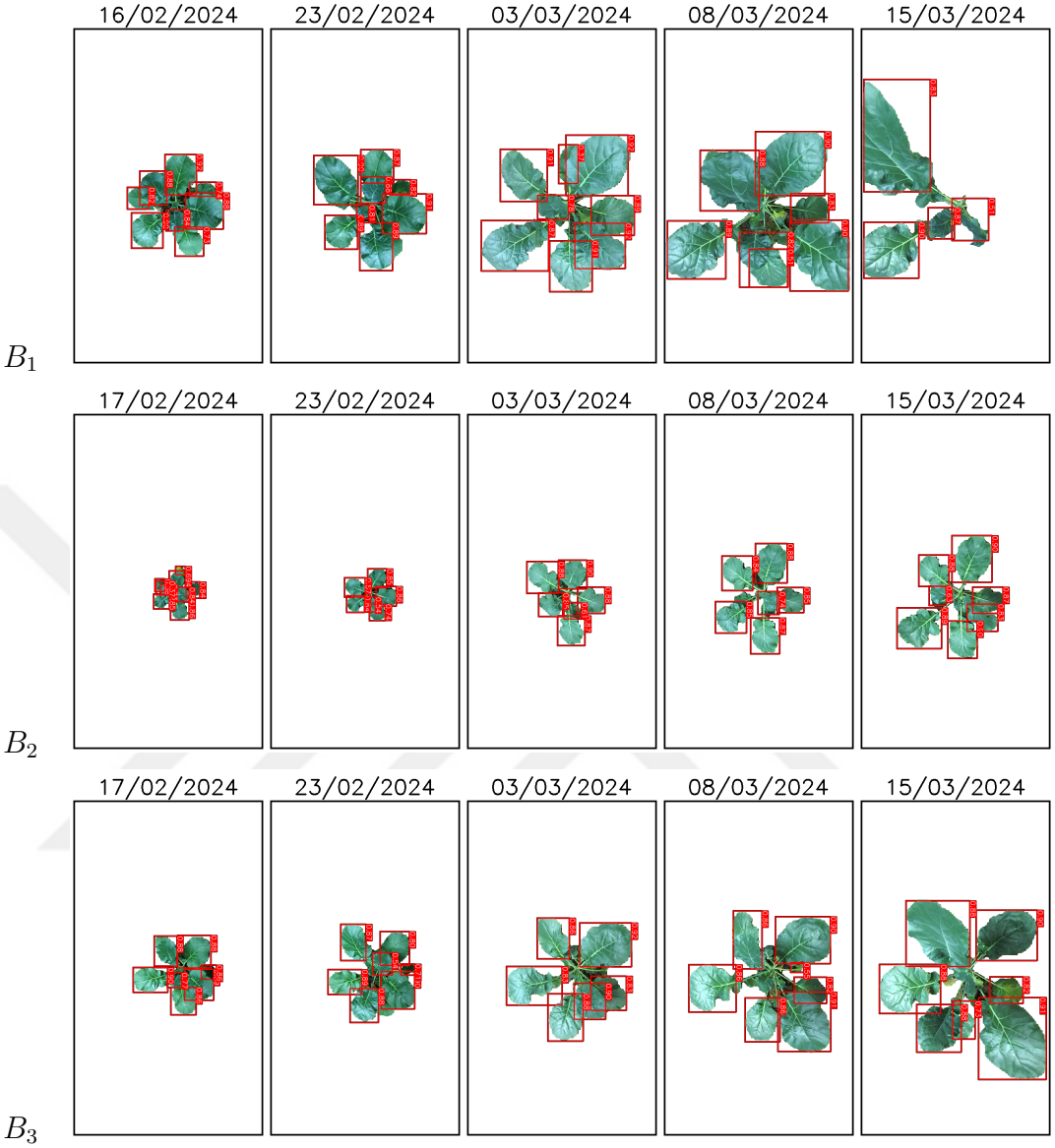


Figure 3.25. Detection of broccoli leaves vs time.

Figure 3.25 shows that the results for broccoli are almost perfect for leaf detection. Since broccolis show their leaves to the camera shooting from above they have distinct stems and plant structures compared to other plants. In fact, broccolis are the easiest scenario for leaf detection. However, new and small leaves that emerge from the center can be confused. Even though the plant detection was not correct in last week's image of the broccoli plant B_1, they are nevertheless accurately detected on the determined segment.

3.5.3. Evolution of Plant & Leaf Detection

It is observed that as the plants develop, the resulting plant and leaf detections change accordingly. As an example, consider the broccoli B_2 as shown in Figure 3.26 through a five-week period. It is observed that:

- In the first week, the plant is fairly small with seven leaves and a center stem as also reflected by the corresponding bounding box and segment.
- In the following two weeks, this structure is preserved while the leaves get larger. This is again encoded by the respective bounding boxes and segments.
- In the fourth week, the size of the plant has increased as reflected by the corresponding bounding box and segment. As the leaves grow, the upper leaves cover the lower leaves. Six leaves are correctly detected. A leaf with only part of it visible was not detected.
- In the final week, the plant has gotten even bigger as also captured by the corresponding bounding box and segment. In this case, the robot detects seven leaves.

Hence, it can be concluded that this system can be used to estimate plant phenotypes.

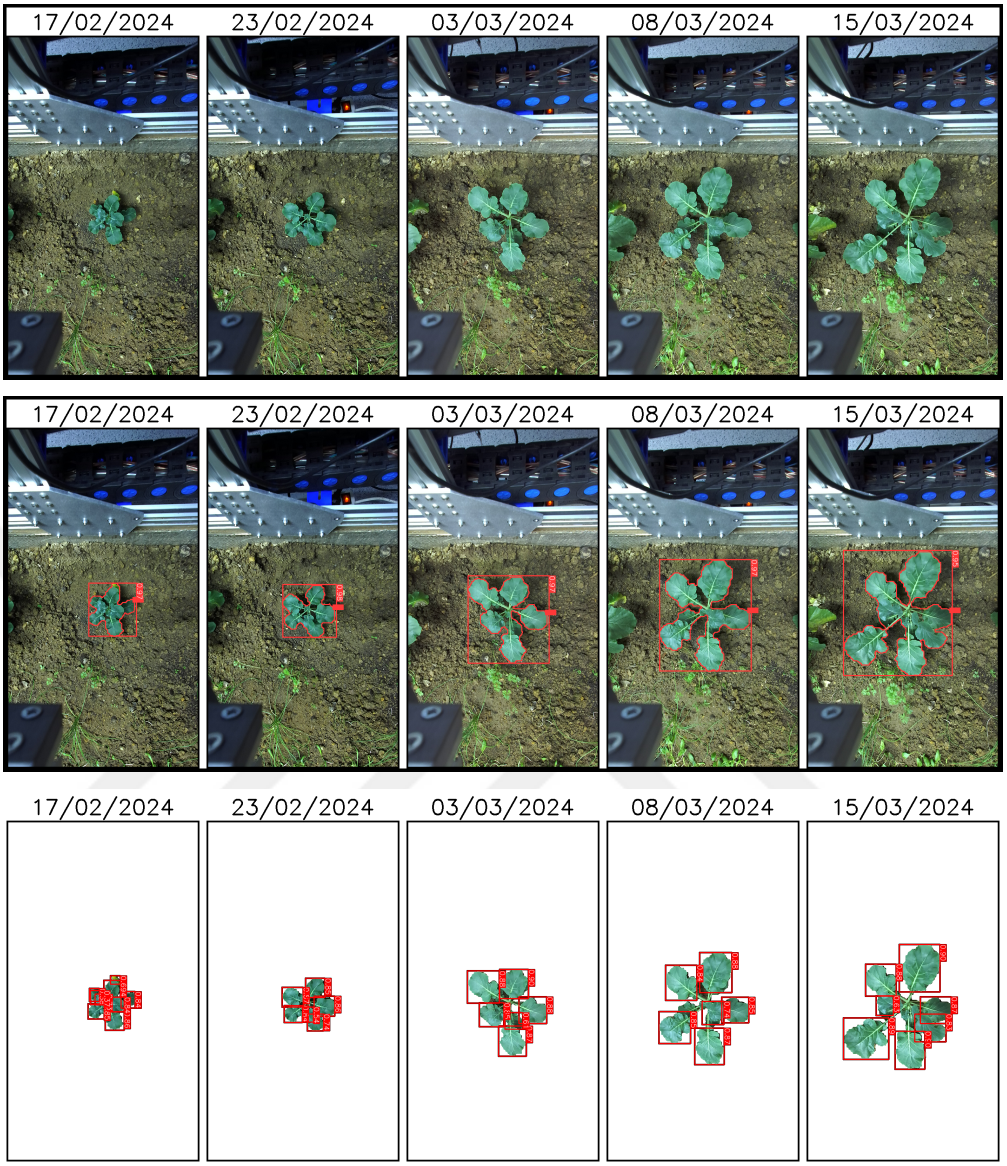


Figure 3.26. Broccoli B_2 plant: images were taken one week apart and plant detection, segmentation, and leaf detection results of RGB-based models on these images, respectively.

4. RGB-D BASED COMPUTATION OF STRUCTURAL PHENOTYPES

This chapter is focused on the computation of the structural phenotypes of individual plants based on visual data [43]. In particular, holistic properties including plant height, plant width, and plant length, as well as the plant's canopy area are considered. In addition, the number of leaves is also determined. The former group are derived from whole-plant geometrics using the point cloud data that is obtained via combining the segmentation mask with the depth data. The number of leaves is a property that is derived from the measurement of individual parts based on RGB data.

4.1. Related Literature

This problem is being increasingly studied in the literature [44]. As non-invasive methods are preferred, sensor-based methods are being developed as surveyed in [45]. For example, RGB data is used to obtain leaf counts in [46] or RGB images obtained through viewing the plant from its side is used to obtain its skeleton [47]. However, such methods cannot be used for three-dimensional models of the plants. Multiview RGB images are used to generate a three-dimensional mesh-based model [48]. A laser scanner mounted on a robot arm is used to capture views of the plant from multiple views which are then used to construct the three-dimensional model of the plant [49]. However, in all these works, the setting consists of a single plant. In some settings, the plants need to be moved to a special chamber for their measurements. Such settings are not realistic for garden or field farming.

Plant traits are also estimated by generating synthetic images [50]. Synthetic plant images are compared with real plant images. Similar to our study, calculate bounding box width and length in pixels, leaf number, and pixel area, and compare these features with those in synthetic images. Unlike our study, the features are not three-dimensional.

4.2. Structural Phenotypes

The structural phenotypes of each plant are obtained based on its bounding box and segment as well as the associated depth data. In particular, geometric properties are derived from constructing its point cloud representation. The mask is applied to both RGB and depth images, keeping only the area corresponding to the plant and removing other parts. A point cloud is then created from this masked data. In particular, depth data is masked by the segmentation mask. This data is then transformed using the camera's intrinsic parameters. For visualization purposes, RGB-D point clouds are used. However, for computational efficiency, only depth-based point cloud representation is utilized.

Width and length calculations rely on the respective segmentation mask. The plant's maximum length and width are computed from its point cloud by finding the differences between the minimum and maximum x and y coordinate values respectively. Height is computed using the bounding box. In this case, depth data is masked by the bounding box. The height is then defined by the difference between the minimum and maximum depth values. The measurement is more accurate when taken closer to the center of the plant. Uneven soil surfaces can cause differences of a few centimeters. However, small differences are acceptable as long as the depth data is accurate and the value used is not corrupted. The segmentation mask does not allow for an accurate height measurement because upper leaves frequently obstruct the view of the plant's root.

Finally, the plant area A_p as viewed orthographically is estimated. In this case, both the bounding box mask and segmentation masks are utilized. The area of the plant is found by finding the percentage of the bounding box that overlaps with the segment [26]. Namely, if A_b refers to the bounding box area, N_s refers to the number of pixels in the segment and N_b refers to the number of pixels in the bounding box,

then

$$A_p = \frac{N_s}{N_b} \cdot A_b \tag{4.1}$$

where A_l is the area of the plant, p_l is the number of pixels in the mask, p_s is the total number of pixels in the bounding box, and A_s is the area calculated from the width and length values.

4.3. Structural Phenotype Computation Pipeline

An end-to-end pipeline was developed for estimating structural phenotypes. This pipeline enables their computation from the RGB-D images collected by the robot according to its daily schedule.

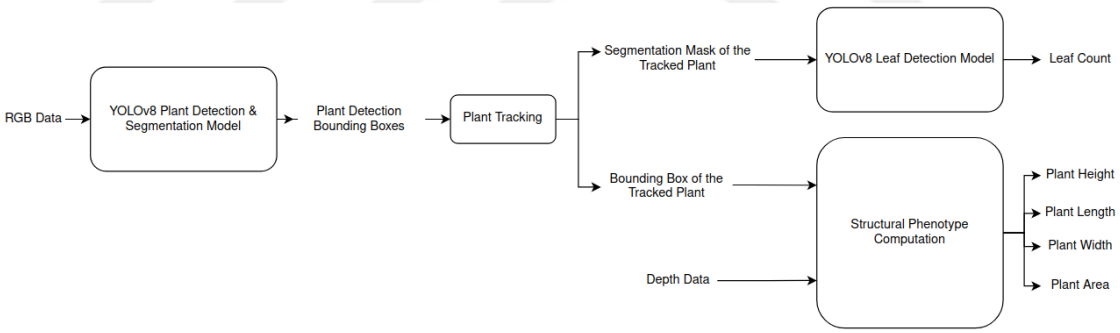


Figure 4.1. Overall structural phenotype computation pipeline.

Figure 4.1 illustrates the blocks that compute plant phenotypes by combining deep learning models with depth data and the outputs of YOLO models. The pipeline is triggered whenever TarRob records a new image. The RGB image is input into the YOLOv8 plant detection & segmentation model as explained in Chapter 3. The YOLOv8 leaf detection model waits for the plant segmentation mask. Upon receiving the mask, it combines the RGB image and segmentation mask to create an image containing only the plant, on which it performs leaf detection. The structural phenotype computation block, represented in the system, takes the depth data recorded by TarRob. It waits for the plant detection bounding box and plant segmentation mask.

Subsequently, one function takes the plant detection bounding box and depth as input and outputs the plant height. Another function, taking the plant segmentation mask and depth data as input, outputs the plant length and width. Using all three inputs, it also outputs the plant area.

4.4. Experimental Results

This section presents the experimental results regarding the computation of structural phenotypes. First, the nature of depth data is discussed as it is critical to the construction of point cloud data associated with each detected plant. Following this, the accuracy of plant height measurements is investigated. Finally, the reliability of the estimated structural phenotypes of individual plants is studied.

4.4.1. Depth Data

It should be noted that the camera's minimum depth reading is 30cm while the maximum depth reading is 20m. The error rate is below 1% for distances up to 3 meters. For greater distances, the error margin may increase up to 5%. However, due to the mechanical design of the robot, the possible maximum reading is 55 cm. This results in an error margin of 3-5mm. Additionally, homogeneous and textureless surfaces, as well as overexposed areas, can increase measurement errors. These conditions negatively affect the matching algorithms in stereo cameras [51].

Before measurements, the OpenCV undistortion method is applied using lens distortion coefficients as given in Appendix A.3. Following, various filtering methods are utilized in order to minimize the number of image points with missing depth data: Gaussian, bilinear, and FFT interpolation. All have proven to be unsatisfactory they lead to unwanted blurred points between missing data and the real depth values on the depth image.

The median filter and closing methods perform better. The closing filter provided the most points on the point cloud. Although this method is prone to over-smoothing and loss of resolution, it outperforms the median filter. The median filter, while effective, is computationally costly and cannot close gaps as effectively as the closing method. Finally, the closing method with a kernel size of 5 was added to the measurement nodes, balancing the need for gap closure with minimizing resolution loss.

4.4.2. Accuracy of Geometric Computations

The accuracy of height measurements is studied in several scenarios. As explained, the error rate of the camera is 1%. An error of 5 mm would be a normal result. Moreover, an error value under 0.5 cm is quite accurate for the plant phenotype estimation task.

First, four wooden blocks with known heights were stacked on top of each other one at a time, and the distance was measured at each stage. As the texture of the surface can affect the measurement of the camera, the upper block is kept the same throughout the measurements.



Figure 4.2. Wooden blocks setup and computed surface.

As an example, Figure 4.2 shows an image of three wooden blocks stacked on top of each other and the corresponding surface in the resulting point cloud. It is observed that the depth reading from a flat surface tends to be noisy so the height readings are not the same across all the surface points. Rather, they tend to vary slightly from each other. The distance to the surface of stacked blocks is measured for different numbers

of stacked blocks as n_{stack}. The computed depth for each n_{stack} is shown in Table 4.1.

Table 4.1. Depth measurement of stacked wooden blocks.

n_{stack}	**Computed Depth (cm)**	**Error (mm)**
1	50.313	± 1.23
2	48.182	± 1.26
3	46.172	± 1.45
4	44.107	± 1.58

To estimate the height of a single block, we calculate the difference between consecutive depth measurements as follows

$$h_{\text{comp}} = d_{n_{\text{stack}}} - d_{n_{\text{stack}}+1} \tag{4.2}$$

where h_{comp} is the computed height of a single block, $d_{n_{\text{stack}}}$ is the computed depth for n_{stack} stacked blocks, and $d_{n_{\text{stack}}+1}$ is the computed depth for $n_{\text{stack}} + 1$ stacked blocks.

The results of these computations for three blocks are presented in Table 4.2, where they are compared with the actual measured height of 2.2 cm.

Table 4.2. Height computation of individual wooden blocks.

Block ID	**Measured Height (cm)**	**Computed Height (cm)**	**Error (mm)**
1	2.2	2.131	± 2.49
2		2.010	± 2.71
3		2.065	± 2.81



Figure 4.3. Width and length measurement of the cardboard box.

In the next scenario, since it's impossible for the soil surface to be perfectly flat, a cardboard box was placed on it and confirmed to be parallel to the ground using a spirit level as seen in Figure 4.3. The cardboard box ground was used to create a flat surface, as the soil is rugged and the height varies across the area. The images were cropped to include only the area of the cardboard and combined with depth data for accurate measurements. The length and width dimensions of a cardboard box were calculated using point cloud data obtained from RGB-D data and compared with manually made measurements as seen in Table 4.3. The error value in measurements made with point cloud data is below 2 mm. Similar to the results of previous verification tests, this value is below the 1% limit predicted by ZED 2.

Table 4.3. Cardboard box scenario.

	Measured (cm)	**Computed (cm)**
Width	6.4	6.466
Length	14.2	14.178
Height	8.3	8.495

4.4.3. Structural Phenotype Measurements

In these experiments, structural phenotype values are computed autonomously. Once a plant is detected and segmented, the resulting segmentation mask is used to

derive point cloud data from the RGB-D data. Following, all structural phenotypes are computed.



Figure 4.4. Measurement verification for cardboard box.

Figure 4.4 shows a leaf on a stick placed on a cardboard box surface. The plant detector's confidence threshold was set to 0.05 to detect only one leaf as a plant.



Figure 4.5. Measurement verification for the whole system.

Figure 4.5 illustrates the bounding box created by the plant detector, segmentation mask area, and resulting point cloud. Actual measurements are compared with those computed as seen in Table 4.4. The error in measurement results is less than 4 mm. The error rate below the 1% rule was maintained in the measurement from approximately 40 cm. In addition, unlike wooden blocks, the plant leaf does not have a fixed shape, and small changes in its shape when left on the surface may increase the error value.

Table 4.4. Leaf instance's dimensions.

	Measured (cm)	**Computed (cm)**
Height	10.0	10.122
Width	11.2	11.584
Length	7.2	7.557

4.4.4. Structural Phenotype Measurements on Various Plant Species

The structural phenotype computation pipeline applies plant detection, segmentation, and leaf detection sequentially to the RGB-D data of plants. It combines the segmentation masks and bounding box data from deep learning models with depth data to measure the height, width, and length of plants. The stage that uses depth data to measure plant dimensions is applicable on all plant species. This is due to the fact the plant detection and segmentation model was trained on various plants from the TarRob dataset, and the leaf detection model was trained on diverse leaf images from both an external dataset and the TarRob dataset. Consequently, the pipeline can be applied to plant species that have not previously been encountered by the models.



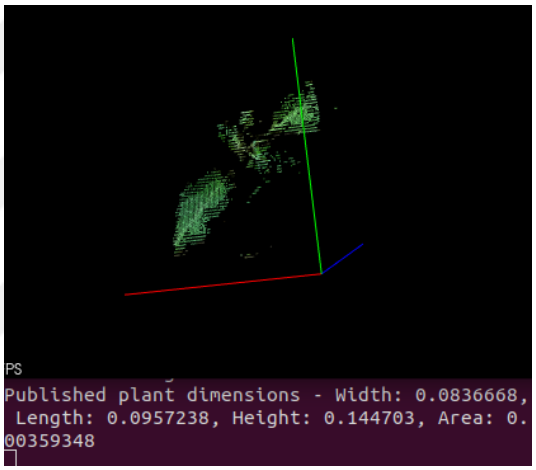
(a) Original image of the pepper.



(b) Plant detection and segmentation results on pepper image.



(c) Leaf detection results on pepper image.



(d) Pcl measurement results on pepper image.

Figure 4.6. Structural phenotype computation pipeline stages on pepper image.

For this, four new seedlings are planted: two pepper and two tomato seedlings. Figure 4.6 demonstrates the results obtained at each stage of the structural phenotype computation pipeline for the first pepper plant as seen in Figure 4.6(a). Figure 4.6(b) shows the output of the YOLOv8 plant detection and segmentation model. Despite never having encountered a pepper plant before, the model successfully detected the plant with a high confidence rate of 0.93 and performed effective segmentation. Note that the relative simplicity of the scene with the plants being easily distinguishable facilitates the model's accurate prediction.

Figure 4.6(c) illustrates the leaf detection results on the segmented part of the original image, which was used as input for the leaf detection model. The model

estimated four leaves, which is consistent with the visible leaves. However, it failed to detect two small leaves on the plant, missing the actual total number of six leaves. This error is due to the model's lower recall value for newly emerged small leaves, a trade-off to maintain precision in leaf detection.

Figure 4.6(d) displays the PCL data of the pepper plant and the height, length, and width features calculated using this data. Table 4.5 compares values obtained through manual measurement and the automated pipeline. It should be noted that manual measurement is not a high-precision method. Errors of a few centimeters in width and length measurements of plants are generally acceptable. Unlike the wooden blocks used in previous verification, plants do not have fixed and definite shapes.

Table 4.5. Dimensions of the pepper.

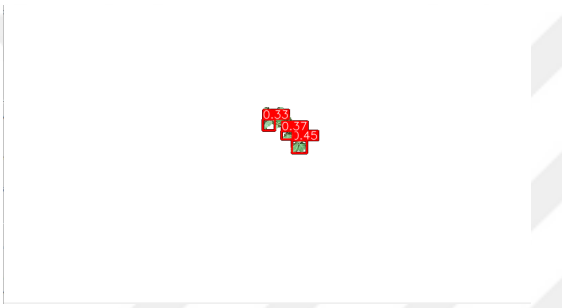
	Measured (cm)	**Computed (cm)**
Height	14.0	14.470
Width	9.0	8.367
Length	9.0	9.572



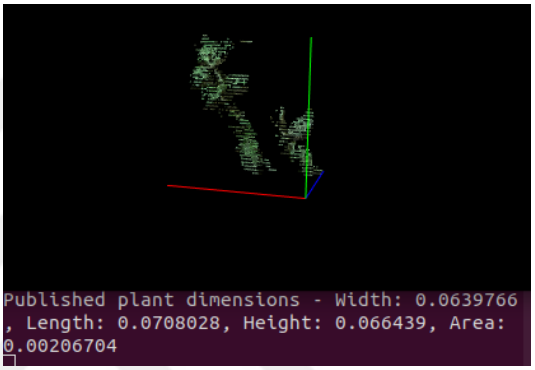
(a) Original image of the tomatoes.



(b) Plant detection and segmentation results on tomatoes image.



(c) Leaf detection results on tomatoes image.



(d) Pcl measurement results on tomatoes image.

Figure 4.7. Structural phenotype computation pipeline stages on tomatoes image.

Figure 4.7 illustrates the results of each stage in the pipeline for a tomato plant. The tomato plant and its plant detection and segmentation results are as seen in Figure 4.7(a) and Figure 4.7(b). Similar to the pepper plant, the model achieved a high confidence rate and accurate prediction for the tomato. Again, the fact that the seedling is planted in a vacant area enhances the model's detection confidence.

Figure 4.7(c) shows the leaf detection results. The leaf detection model predicted three leaves with low confidence rates. With a careful examination, thirteen leaves are visible in the original image. While some leaves are challenging to detect, the leaf detection model's performance on the tomato plant is not optimal. Possible reasons include the model not being fine-tuned for these specific leaf types and the plant and its leaves being significantly smaller than what the model usually encounters. Figure 4.7(d) presents the plant's PCL data and measurement results. The model in Figure 4.7(b)

was observed to produce a segmentation mask that was smaller than the actual plant area. This is attributed to the segmentation mask being slightly smaller than the actual plant area. Therefore computed dimensions are smaller than the manual measurements as seen in Table 4.6

Table 4.6. Dimensions of the tomatoes.

	Measured (cm)	**Computed (cm)**
Height	6.5	6.644
Width	7.5	6.398
Length	7.5	7.080



(a) Original image of the beet.

(b) Plant detection and segmentation results on beet image.

Figure 4.8. Structural phenotype computation pipeline stages on beet image.

Finally, performance is also evaluated on a beet plant. This is a plant that is fairly mature in its growth. The beet plant and its plant detection and segmentation results are seen in Figure 4.8(a) and Figure 4.8(b). The plant detection model failed to detect the beet. Instead, the chard was detected in the right corner of the scene. Interestingly, even when decreasing the confidence threshold, beet detection and segmentation do not occur.

The primary reason for this is the model's inability to predict the shape of a plant it has never encountered before within a complex scene. Figure 4.8 demonstrates a case where the model did not detect the plant. This may be because plant detection

models are trained with a dataset annotated with bounding boxes that include the entire plant. It does not detect a few branches and leaves as a plant with the highest F1 score threshold. It can only be detected with a low confidence rate. Even in this scenario, it detected chard leaves it had seen before. At the lowest confidence rates, instead of finding the shape of a whole plant, it detects all parts that are different from the background and resemble the plant. The simplest solution would be to train the model specifically on beet plants. Alternatively, using larger models trained on extensive datasets could improve overall plant detection accuracy and the model's intuition of plant shapes.

5. LEARNING AND PREDICTING PLANT GROWTH

Tracking the structural phenotypes on a temporal basis is integral to the learning and prediction of plant growth. As the plant develops, structural phenotypes are associated with quantifiable changes over time and are thus reliable indicators of plants' growth status. As such, the temporal progress of the phenotypes computed for each plant species forms a foundation for plant growth monitoring during each farming season.

5.1. Related Literature

Monitoring plant growth efficiently is an integral aspect of farming production and planning [11]. By combining the plant's structural traits data with sensor data such as humidity, light, and temperature, the relationship between plant growth and external factors can be learned [52]. Plant growth monitoring requires the use of temporal data. In many works, the growth period is discretized into several stages and then growth is tracked by finding the current growth stage of the plants. For example, four stages (Soil, FA, OC, and FL) are defined in [53]. This is then posed as a classification problem.

Recently, the associated classification problems have increasingly been solved using deep learning networks. Two primary architectures are being used: the Pure CNN network and the hybrid CNN-LSTM network [12]. In the former, the input data is commonly in the form of grayscale images or color 2-D images, providing only the spatial information [54, 55]. However, in the latter, a series of images across a period of time are input, and thus both spatial and temporal information are provided [56].

The biggest dilemma of deep learning methods is their need for large datasets. As such, training may overfit and end in the first epoch. This can be alleviated to some degree by choosing the model that fits the data in the slowest manner [57]. Therefore,

in scenarios with small datasets, if there is not enough data for ANN and ML training, classical statistical methods may surpass learning algorithms [58]. Alternatively, data augmentation methods are used to prevent this [59]. The difficulty of data generation is exacerbated by the fact that as these methods are all supervised, the ground truth references that are required for training need to be obtained from field experts. As such, we are interested in learning the growth trajectory of each phenotype as a time function and hence learning to monitor the plants accordingly.

5.2. Evolution of Structural Phenotypes

The evolution of the structural phenotypes for each plant species is tracked through nearly a two-month period. For each plant, the computed values are recorded between 08/02/2024 and 01/04/2024. The resulting trajectories are obtained as a function of recording times. Hence, for each plant and each phenotype, there are about 200 data points. The gaps in the figures are due to the problematic intervals as discussed in Section 3.4.3. Unfortunately, once the plant height reaches the 20-25 cm range, the accuracy of the computed values decreases. As discussed previously, the camera of the robot can maximally go up to 44 cm above the ground. Due to this, as the plants' height reaches around 25 cm, the reliability of the depth data decreases.

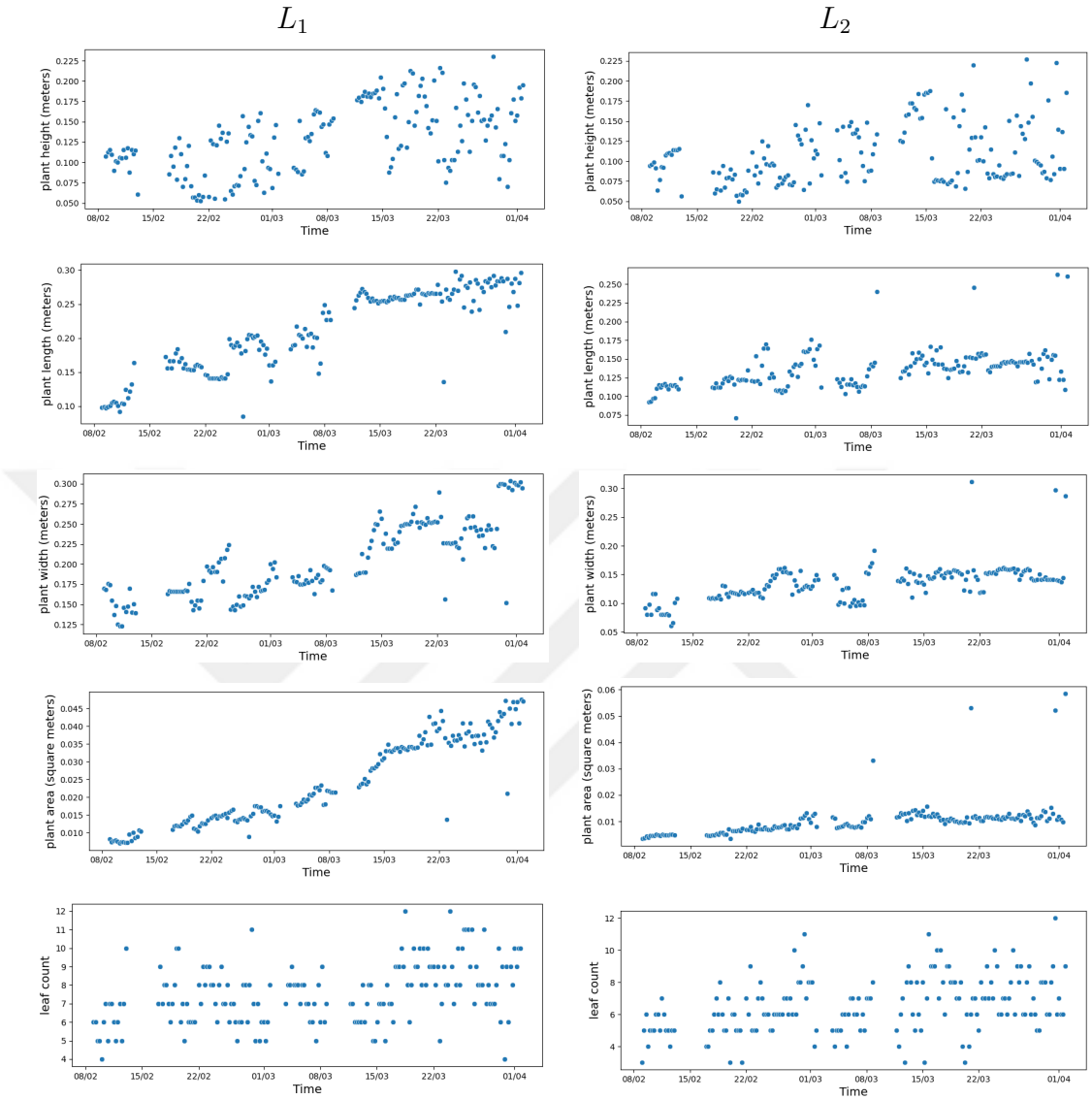


Figure 5.1. Temporal evolution of structural phenotypes - lettuces L_1 and L_2.

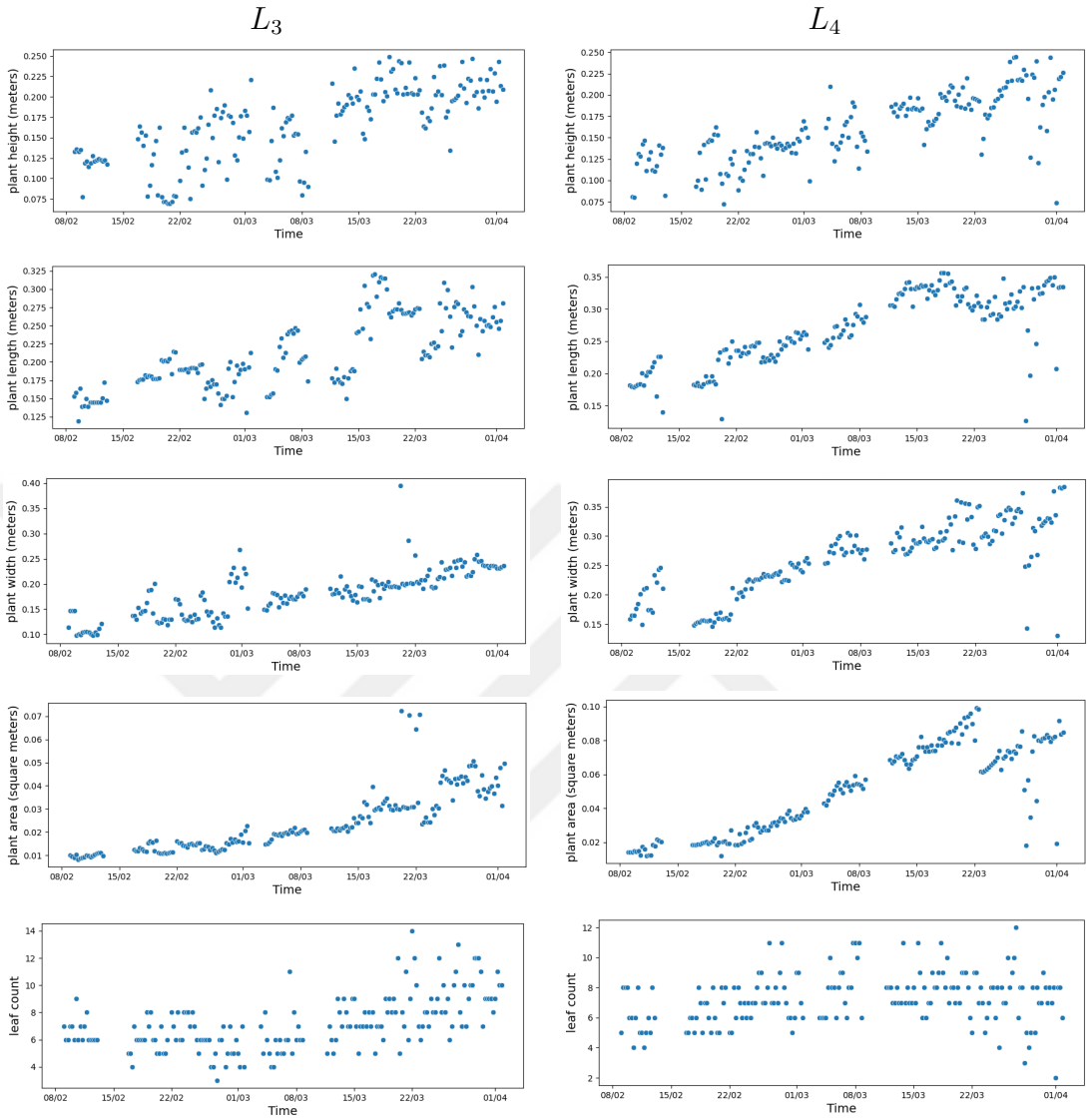


Figure 5.2. Temporal evolution of structural phenotypes - lettuces L_3 and L_4.

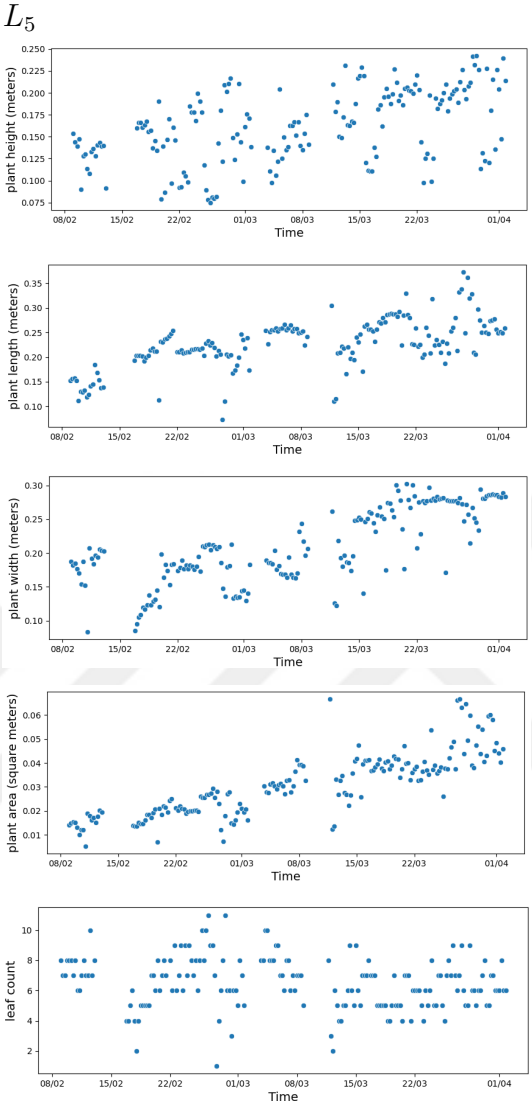


Figure 5.3. Temporal evolution of structural phenotypes - lettuces L_5.

Figure 5.1, Figure 5.2, and Figure 5.3 demonstrate the temporal evolution of structural phenotypes for lettuces in the polyculture garden. The lettuce data exhibits higher variance compared to the chard and broccoli data. This is particularly evident in the plant's height plot - even though the visual detection of this plant is the least affected by the robot's limited workspace height. This may be attributed to:

- Lettuces being planted close to each other and chards.
- Lettuces tend to intrude on each other's area which then reduces detection and segmentation accuracy.

- Rapid height growth of nearby chards with broad leaves obstruct the camera view of lettuces.

Additionally, unlike broccoli and chard, lettuces lack strong stems. Consequently, taller leaves can easily flatten due to watering or interference from other plants' leaves, potentially decreasing plant height measurements.

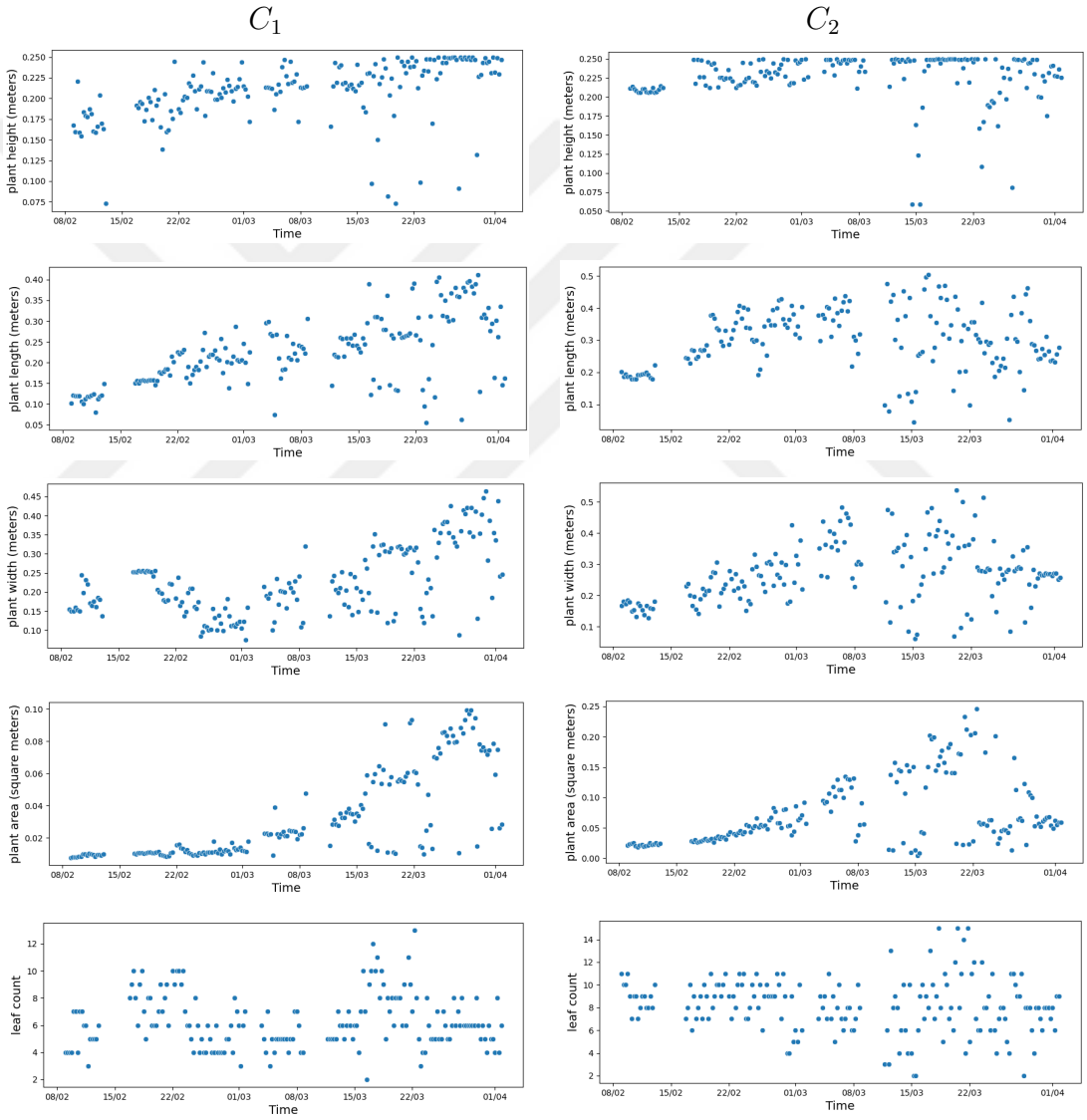


Figure 5.4. Temporal evolution of structural phenotypes - chards C_1 and C_2.

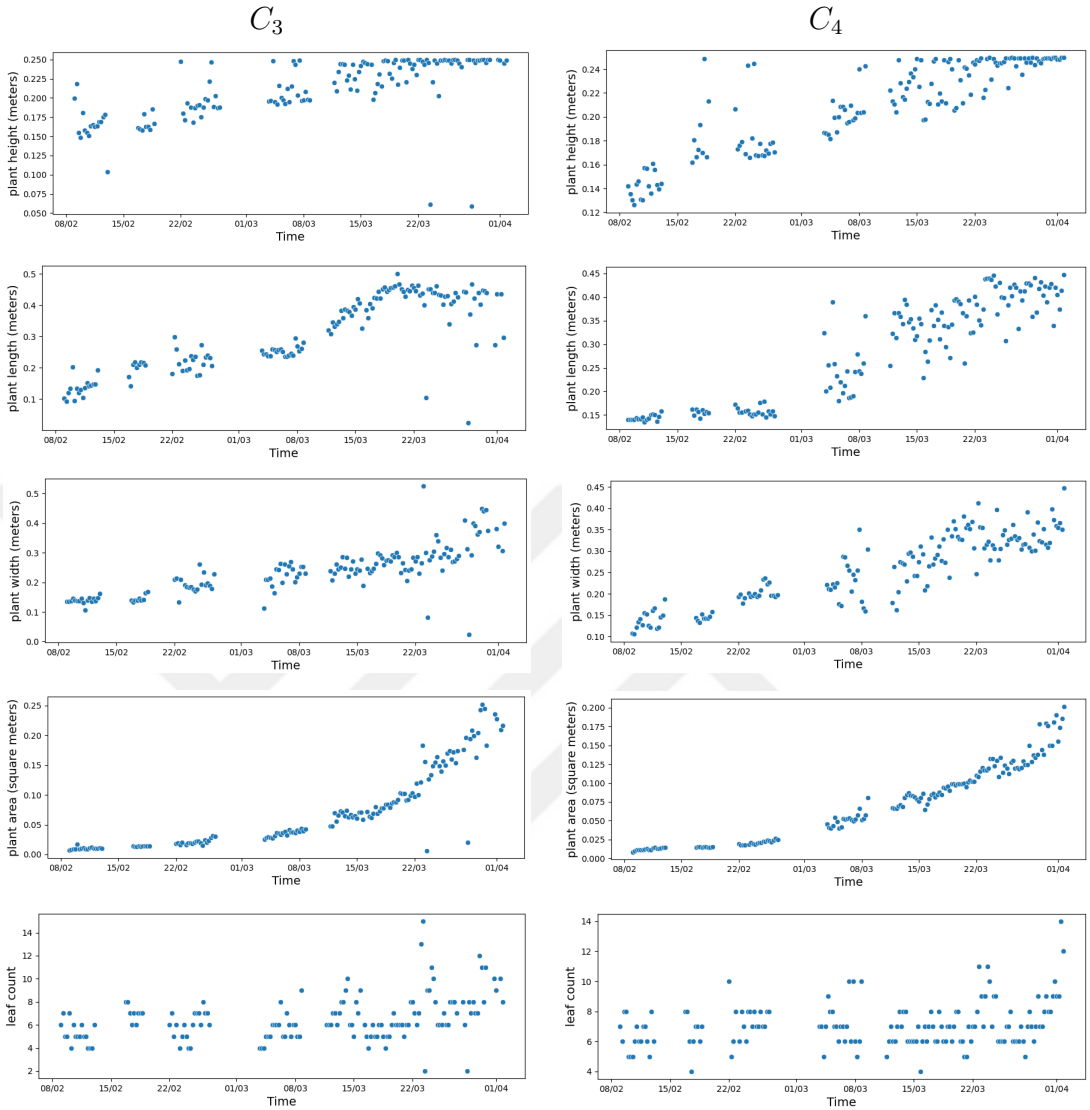


Figure 5.5. Temporal evolution of structural phenotypes - chards C_3 and C_4.

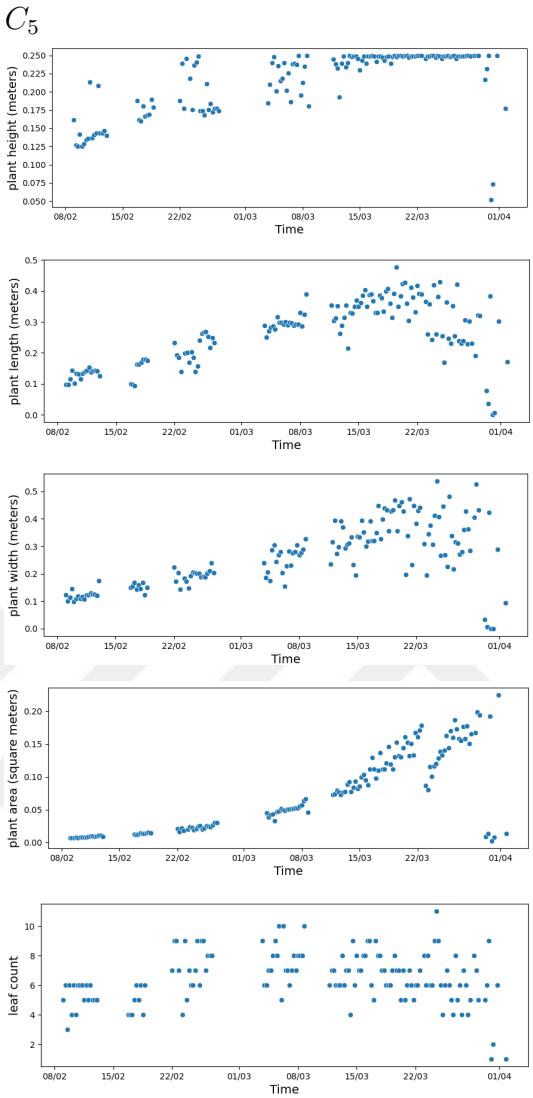


Figure 5.6. Temporal evolution of structural phenotypes - chard plant C_5.

Figure 5.4, Figure 5.5, and Figure 5.6 demonstrate the temporal evolution of structural phenotypes for chards in the polyculture garden. Chards are tall, fast-growing plants. The robot setup's height most significantly affected chard measurements. After plant height exceeded 20 cm, the error rate and variance in measurements increased substantially. For instance, the variance in C_5's temporal data rises from March 15 onwards. This date coincides with the plant reaching the 25 cm threshold. The leaf count phenotype of chards is more stationary compared to lettuces. Among broccoli, chard, and lettuce plants, lettuce yields the least reliable leaf detection model results due to its lack of stems and overlapping leaves. Although the accuracy of chard measurements decreases once its height reaches 25 cm, growth trends can still

be observed through the movement of the mean value of the plant traits.

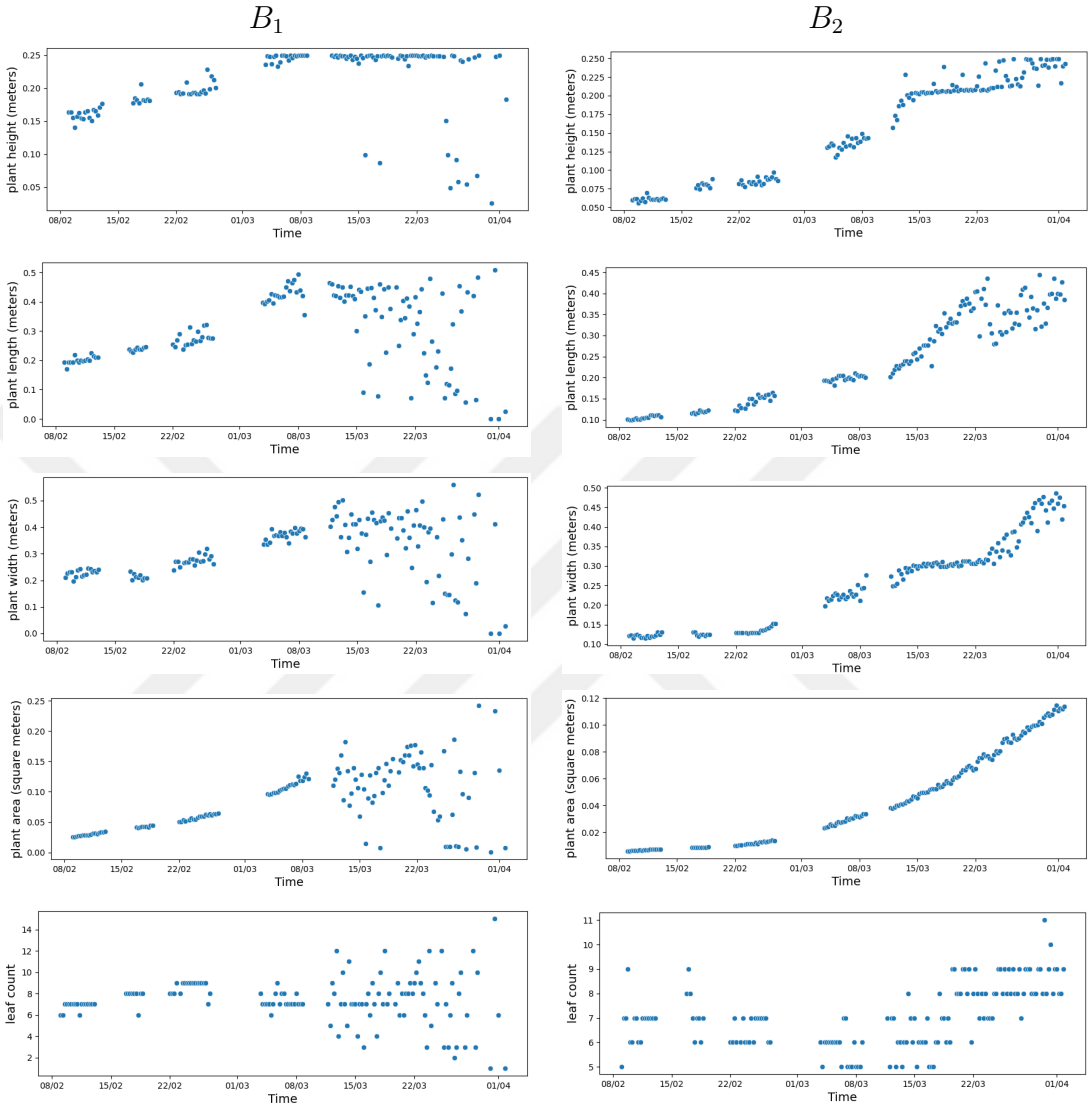


Figure 5.7. Temporal evolution of structural phenotypes - broccolis B_1 and B_2.

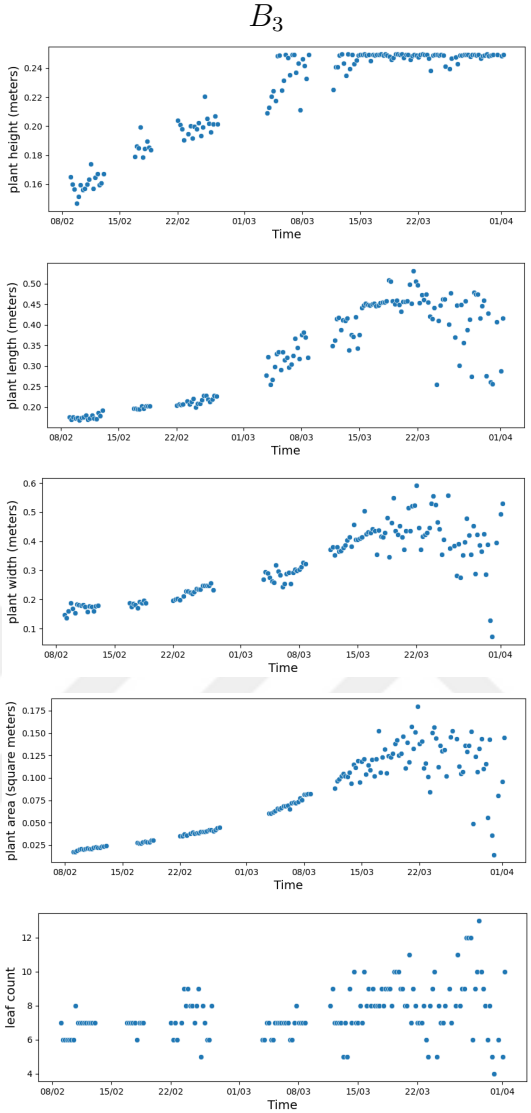


Figure 5.8. Temporal evolution of structural phenotypes - broccoli plant B_3.

Figure 5.7 and Figure 5.8 demonstrate the temporal evolution of structural phenotypes for broccolis. Broccoli is the plant with the most reliable estimated phenotype values. This is attributed to broccolis being planted at the recommended distance from the other plants, and having strong stems that enable their shape and height to be stable. Their leaves are easier to detect. Let it be noted that the temporal data of the broccolis also demonstrates issues pertaining to the size-wise limitation of the robot's workspace.

Leaf count is the most unpredictable structural phenotype. Chard or lettuce measurements can easily fluctuate due to influences from water, light, or neighboring plants' leaves. While the actual leaf count may remain constant, the number of detected leaves varies continuously in measurements. In contrast, broccoli data shows consistent leaf detection across consecutive measurements. Additionally, broccoli graphs exhibit a noticeable change in their variance when the plant reaches its maximum measurable length. Unlike chard, when broccoli reaches its maximum plant height, all of its features lose accuracy. This is because broccoli is morphologically a broad plant and its canopy area goes out of the bound of the camera's field of view.

5.3. Statistical Forecasting on Time-Series

In the context of plant growth, statistical forecasting methods can provide valuable insights into future growth trends and help in optimizing cultivation practices. Among various time series models, the ARIMA (AutoRegressive Integrated Moving Average) model is a popular method for time series forecasting and has proven to be particularly effective and easy to implement. This is a univariate time series model. Fifteen ARIMA models are required to analyze three plant species with five plant phenotypes each.

5.3.1. Estimating ARIMA Parameters

The ARIMA model is defined by three parameters [60].

- AutoRegressive order p: Number of lag observations included in the model.
- Differencing order d: Number of times the raw observations are differenced to make the time series stationary.
- Moving Average order q : Number of lagged forecast errors in the prediction equation.

The bayesian information criterion (BIC) is used to evaluate model performance, balancing model fit with complexity [61]. Models with higher p and q parameters are prone to overfitting but more complex. Table 5.1 shows the optimal ARIMA parameters determined by BIC for different plant features across lettuce, chard, and broccoli datasets. The selected p and q parameters for lettuce and chard are nearly identical. For broccoli, lower-order parameters were chosen compared to chard and lettuce. The model selected high lag values for lettuce and chard predictions, attempting to capture long-term relationships in the data.

Table 5.1. ARIMA parameters (p, d, q) for different plant features.

Feature	**Lettuce**	**Chard**	**Broccoli**
Plant Height	(3,1,4)	(3,1,4)	(4,1,4)
Plant Width	(3,1,4)	(3,1,4)	(1,1,2)
Plant Length	(3,1,4)	(3,1,4)	(1,1,2)
Plant Area	(3,1,3)	(3,1,4)	(1,1,2)
Leaf Count	(3,1,4)	(3,1,4)	(1,1,1)

5.3.2. Testing ARIMA Predictions

One sample from each plant species was predicted with ARIMA models using the estimated parameters. Figure 5.9 and Figure 5.10 show the ARIMA model's predictions for three test plants from each of the three species, using species-specific parameters. As previously mentioned, broccoli phenotypes have less variance and are easily predictable data.

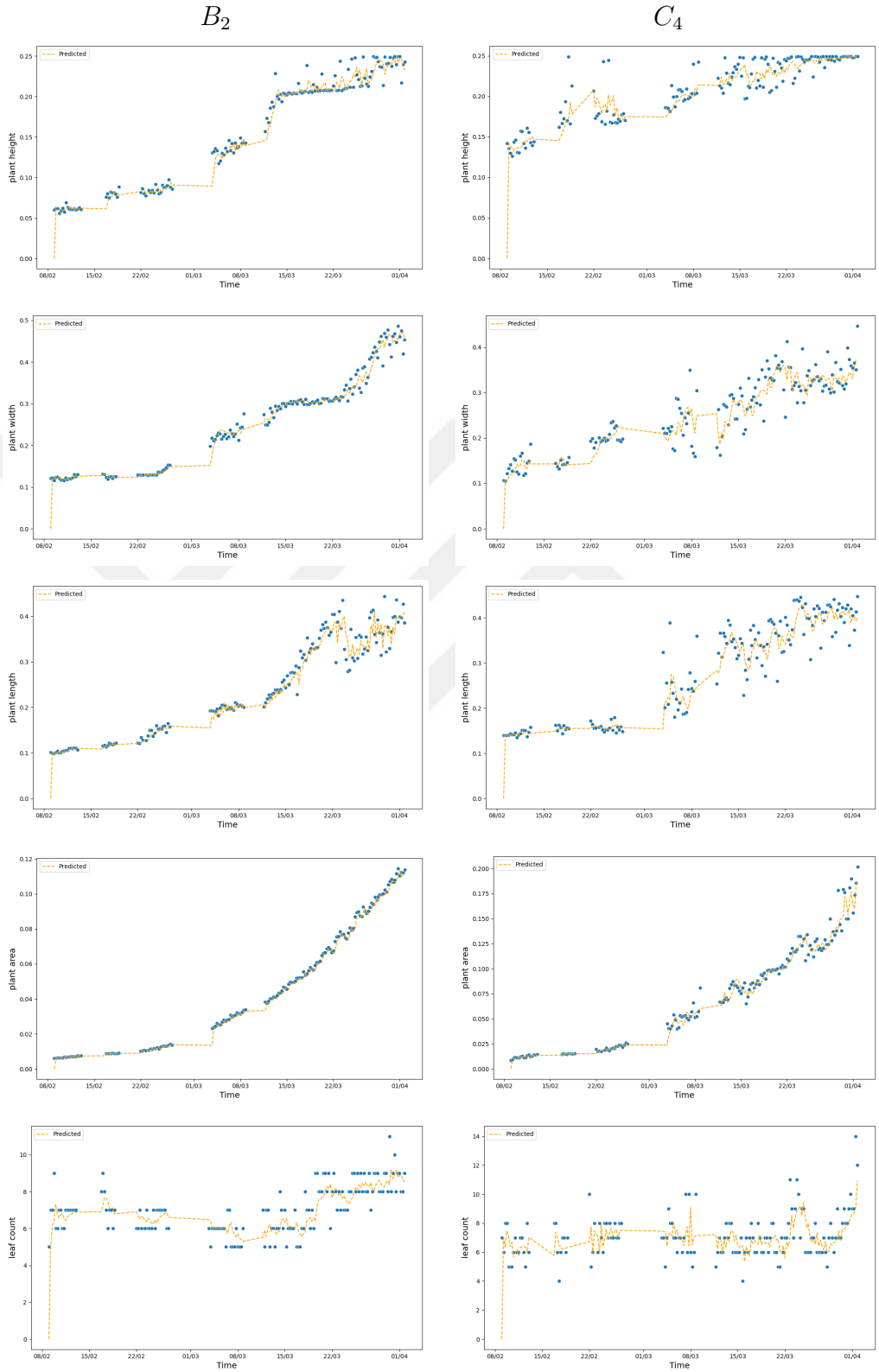


Figure 5.9. ARIMA prediction on structural phenotypes - broccoli B_2 and chard C_4.

L_4



Figure 5.10. ARIMA prediction on structural phenotypes - lettuce L_4.

The prediction accuracy of the ARIMA models is evaluated using the normalized root mean square error (NRMSE) value. This value corresponds to a standardized measure of the prediction error and thus enables the comparison across different plant features and species. It is defined as

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\bar{y}} \tag{5.1}$$

where y_i is the observed value of the plant feature at time i, $\hat{y}_i$ is the predicted value by ARIMA at time i, $\bar{y}$ is the mean of the observed values and N is the total number of observations in the test set. A lower NRMSE indicates better model performance. The NRMSE values for different plant features from the example of the three plant species are presented in Table 5.2. As observed, the model performs better in predicting plant height and width compared to area and leaf count across all plant types.

Table 5.2. NRMSE values for plant features across different plant species.

Plant Species	**Height**	**Width**	**Length**	**Area**	**Leaf Count**
Lettuce	0.138755	0.131249	0.127478	0.106394	0.156671
Chard	0.163584	0.105002	0.131754	0.040531	0.143195
Broccoli	0.063098	0.051234	0.074766	0.020380	0.161444

6. CONCLUSION

This thesis is focused on vision-based robotic plant growth monitoring and learning in polyculture farming. This type of agriculture dictates special soil preparation and farming a variety of companion plants together and hence differs from monoculture agriculture. In this thesis, this problem is addressed in two parts. The first part focuses on the enhancement of the TarRob robot to enable the collection of a polyculture dataset. In the second part, monitoring and learning capabilities of the robot have been developed.

Chapter 2 has presented the development of a polyculture garden and TarRob robot. A new RGB-D TarRob dataset has been constructed. This dataset includes cleaned RGB-D data collected over one and a half months and RGB data spanning two months. Despite challenges in continuous data collection due to simultaneous hardware and software development, a dataset that is suitable for automated plant monitoring has been developed.

Chapter 3 has presented a complete learning-based pipeline for plant detection and segmentation based on RGB data and YOLO models. The experimental results indicate varying accuracy across different plant species. Results were better for broccoli, chard, and lettuce compared to spring onions. Leaf detection performed well for broccoli and chard, with acceptable results for lettuce. These differences in performance are mostly due to the plants' physical characteristics, instead of the model's limitations. For training the plant detection and segmentation models, the TarRob dataset has been expanded to include the ground truth of bounding boxes and segmentation masks.

Chapter 4 has presented novel methods for the computation of structural phenotypes including height, width, length, plant top area, and number of leaves. This includes constructing the point cloud model of each plant and then using this to infer

the structural phenotypes.

Chapter 5 is on the temporal evolution of structural phenotypes to learn individual plant species growth models. These models can then be used to monitor and predict growth during a particular farming season. The temporal evolution of each phenotype of each plant is recorded as time-series data. Although the time-series data was limited for optimal deep learning performance, the models still captured growth patterns to some extent.

As future work, expanding the TarRob dataset with data from future harvests would improve model accuracy, especially for time-series learning. This would provide a larger dataset for learning from more plant instances. Moreover, improving the TarRob system's physical setup is necessary. Reducing plant overlap would enhance detection and segmentation tasks, but this creates a trade-off with maximizing the plant-covered area in limited space. Elevating the system would allow better depth data collection for taller plants, addressing the stereo camera's minimum distance requirement of 30 cm.

Generating synthetic plant data could enhance both RGB and time-series models for more robust deep learning model training. This approach could help overcome the limitations of real-world data collection. As a final improvement, integrating additional sensors like humidity, temperature, and light sensors into the robot system would allow us to study how environmental factors affect plant growth.

REFERENCES

1. Gliessman, S. and M. Altieri, "Polyculture Cropping Has Advantages", *California Agriculture*, Vol. 36, No. 7, pp. 14–16, 1985.

2. Crews, T. E., W. Carton and L. Olsson, "Is the Future of Agriculture Perennial? Imperatives and Opportunities to Reinvent Agriculture by Shifting from Annual Monocultures to Perennial Polycultures", *Global Sustainability*, Vol. 1, No. e11, pp. 1–18, 2018.

3. Mollison, B., *An Introduction to Permaculture*, Yankee Permaculture, Barking Frogs Permaculture Center, Sparr, FL, 2001.

4. Tsaftaris, S. A., M. Minervini and H. Scharr, "Machine Learning for Plant Phenotyping Needs Image Processing", *Trends Plant Sci.*, Vol. 21, No. 12, pp. 989–991, 2016.

5. Birrell, S., J. Hughes, J. Y. Cai and F. Iida, "A Field-tested Robotic Harvesting System for Iceberg Lettuce", *Journal of Field Robotics*, Vol. 37, No. 2, pp. 225–245, 2020.

6. Luo, J., B. Li and C. Leung, "A Survey of Computer Vision Technologies In Urban and Controlled-environment Agriculture", ArXiv:2210.11318, 2023.

7. Ozguven, M., *The Digital Age in Agriculture*, CRC Press, 2023.

8. Tian, H., T. Wang, Y. Liu, X. Qiao and Y. Li, "Computer Vision Technology in Agricultural Automation — A review", *Information Processing in Agriculture*, Vol. 7, No. 1, pp. 1–19, 2020.

9. Kolhar, M., "Plant Trait Estimation and Classification Studies in Plant Phenotyping Using Machine Vision – A Review", *Information Processing in Agriculture*,

Vol. 10, No. 1, pp. 114–135, 2023.

10. Bharman, P., S. Saad, S. Khan, I. Jahan, M. Ray and M. Biswas, "Deep Learning in Agriculture: A Review", *Asian Journal of Computer Science and Information Technology*, Vol. 13, pp. 28–47, 2022.

11. Sharma, R., P. Vaidya and B. Sharma, "A Review on Plant Growth Monitoring using Artificial Intelligence and the Internet of Things", *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 140–145, New Delhi, India, 2024.

12. Tong, Y.-S., T.-H. Lee and K. S. Yen, "Deep Learning for Image-Based Plant Growth Monitoring: A Review", *International Journal of Engineering and Technology Innovation*, Vol. 12, pp. 225–246, 2022.

13. Adebola, S., R. Parikh, M. Presten, S. Sharma, S. Aeron, A. Rao, S. Mukherjee, T. Qu, C. Wistrom, E. Solowjow and K. Goldberg, "Can Machines Garden? Systematically Comparing the AlphaGarden vs. Professional Horticulturalists", *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

14. "FarmBot", `https://farm.bot`, accessed on July 22, 2023.

15. Doksansoglu, H. I., *CNC Robot for Indoor Farming*, Tech. rep., EE 492 Senior Project Report, Electrical and Electronics Engineering, Bogazici University, 2017.

16. Dogan, M. D. and U. Soylu, *CNC-Based Intelligent System for Automated Indoor Farming*, Tech. rep., EE 492 Senior Project Report, Electrical and Electronics Engineering, Bogazici University, 2017-2018.

17. Takım, M. B. and c. B. Çakır, *Intelligent Autonomous Robot for Precise Indoor Farming*, Tech. rep., EE 492 Senior Project Report, Electrical and Electronics Engineering, Bogazici University, 2019.

18. Tisné, S., Y. Serrand, L. Bach, E. Gilbault, R. Ben Ameur, H. Balasse, R. Voisin, D. Bouchez, M. Durand-Tardif, P. Guerche, G. Chareyron, J. Da Rugna, C. Camilleri and O. Loudet, "Phenoscope: An Automated Large-scale Phenotyping Platform Offering High Spatial Homogeneity", *The Plant Journal*, Vol. 74, No. 3, pp. 534–544, 2013.

19. Hemenway, T., *Gaia's Garden: A Guide to Home-Scale Permaculture*, Chelsea Green Publishing, White River Junction, VT, 2009.

20. Mollison, B. and R. M. Slay, *Introduction to Permaculture*, Tagari Publications, 1997.

21. Salunkhe, D. K. and S. S. Kadam, *Handbook of Vegetable Science and Technology: Production, Composition, Storage, and Processing*, CRC Press, Boca Raton, 1998.

22. Schuh, M. and J. MacKenzie, "Growing Spinach and Swiss Chard in Home Gardens", 2022, `https://extension.umn.edu/vegetables/growing-spinach-and-swiss\\-chard#soil-testing-and-fertilizer-389360`, accessed on October 08, 2023.

23. Drost, D., "How to Grow Swiss Chard in Your Garden", 2020, `https://extension.usu.edu/yardandgarden/research/swiss-chard-in-the-\\garden`, accessed on October 08, 2023.

24. Steil, A., "Vegetable Harvest Guide", 2023, `https://yardandgarden.extension.iastate.edu/how-to/vegetable-harvest\\-guide`, accessed on October 08, 2023.

25. Masabni, J., "Texas Home Vegetable Gardening Guide", 2020, `https://agrilifeextension.tamu.edu/library/gardening/texas-home-vege\\table-gardening-guide/`, accessed on October 09, 2023.

26. Chaudhary, P., S. Godara, A. N. Cheeran and A. Chaudhari, "Fast and Accurate

Method for Leaf Area Measurement", *International Journal of Computer Applications*, Vol. 49, 2012.

27. Okayasu, T. *et al.*, "Affordable Field Environmental Monitoring and Plant Growth Measurement System for Smart Agriculture", *2017 Eleventh International Conference on Sensing Technology (ICST)*, Sydney, NSW, Australia, 2017.

28. Bao, Y., L. Tang, M. W. Breitzman, M. G. Salas Fernandez and P. S. Schnable, "Field-based Robotic Phenotyping of Sorghum Plant Architecture using Stereo Vision", *Journal of Field Robotics*, Vol. 36, No. 2, pp. 397–415, 2019.

29. Ward, I. R., H. Laga and M. Bennamoun, "RGB-D Image-Based Object Detection: From Traditional Methods to Deep Learning Techniques", *RGB-D Image Analysis and Processing*, Cham, Switzerland, 2019.

30. Zhang, L., Z. Xu, D. Xu, J. Ma, Y. Chen and Z. Fu, "Growth Monitoring of Greenhouse Lettuce Based on a Convolutional Neural Network", *Hortic Res*, Vol. 7, p. 124, 2020.

31. Praveen Kumar, J. and S. Domnic, "Rosette Plant Segmentation with Leaf Count using Orthogonal Transform and Deep Convolutional Neural Network", *Machine Vision and Applications*, Vol. 31, No. 1, p. 6, 2020.

32. Ubbens, J., M. Cieslak, P. Prusinkiewicz and I. Stavness, "The Use of Plant Models in Deep Learning: An Application to Leaf Counting in Rosette Plants", *Plant Methods*, Vol. 14, No. 1, p. 6, 2018.

33. Lu, J.-Y., C.-L. Chang and Y.-F. Kuo, "Monitoring Growth Rate of Lettuce Using Deep Convolutional Neural Networks", *2019 ASABE Annual International Meeting*, Boston, Massachusetts, 2019.

34. Bosilj, P., E. Aptoula, T. Duckett and G. Cielniak, "Transfer Learning Between Crop Types for Semantic Segmentation of Crops versus Weeds in Precision Agri-

culture", *Journal of Field Robotics*, Vol. 37, No. 1, pp. 7–19, 2020.

35. Reyes-Yanes, A., P. Martinez and R. Ahmad, "Real-time Growth Rate and Fresh Weight Estimation for Little Gem Romaine Lettuce in Aquaponic Grow Beds", *Computers and Electronics in Agriculture*, Vol. 179, p. 105827, 2020.

36. Weyler, J., F. Magistri, E. Marks, Y. L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss and J. Behley, "PhenoBench: A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2024.

37. Liu, S., Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding Dino: Marrying Dino with Grounded Pre-training for Open-set Object Detection", ArXiv:2303.05499, 2023.

38. Kirillov, A., E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár and R. Girshick, "Segment Anything", ArXiv:2304.02643 [cs.CV], 2023.

39. Ren, T., S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang and L. Zhang, "Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks", ArXiv:2401.14159 [cs.CV], 2024.

40. Yalcin, H., "Visual Analysis of Plant Growth using Transfer Learning", *2021 29th Signal Processing and Communications Applications Conference (SIU)*, Istanbul, Turkey, 2021.

41. IDEA-Research, "Grounded-Segment-Anything", 2023, `https://github.com/IDEA-Research/Grounded-Segment-Anything`, accessed on April 3, 2024.

42. Gallagher, J., "Generate Image Augmentations with Roboflow", 2023, `https://blog.roboflow.com/image-augmentation/`, accessed on May 12, 2024.

43. Das Choudhury, S., A. Samal and T. Awada, "Leveraging Image Analysis for High-Throughput Plant Phenotyping", *Front Plant Sci.*, Vol. 10, 2019.

44. Kolhar, S. and J. Jagtap, "Plant Trait Estimation and Classification Studies in Plant Phenotyping using Machine Vision – A Review", *Information Processing in Agriculture*, Vol. 10, No. 1, pp. 114–135, 2023.

45. Li, L., Q. Zhang and D. Huang, "A Review of Imaging Techniques for Plant Phenotyping", *Sensors*, Vol. 14, No. 11, pp. 20078–20111, 2014.

46. Ubbens, J. R. and I. Stavness, "Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks", *Frontiers in Plant Science*, Vol. 8, 2017.

47. Das Choudhury, S., S. Bashyam, Y. Qiu, A. Samal and T. Awada, "Holistic and Component Plant Phenotyping using Temporal Image Sequence", *Plant Methods*, Vol. 14, No. 35, 2018.

48. Paproki, A., X. Sirault, S. Berry, R. Furbank and J. Fripp, "A Novel Mesh Processing-based Technique for 3D Plant Analysis.", *BMC Plant Biol.*, Vol. 12, No. 63, 2012.

49. Chaudhury, A., C. Ward, A. Talasaz, A. G. Ivanov, N. P. A. Huner, B. Grodzinski, R. V. Patel and J. L. Barron, "Computer Vision Based Autonomous Robotic System for 3D Plant Growth Measurement", *2015 12th Conference on Computer and Robot Vision*, pp. 290–296, Halifax, NS, Canada, 2015.

50. Wang, C., W. Pan, X. Song, H. Yu, J. Zhu, P. Liu and X. Li, "Predicting Plant Growth and Development Using Time-Series Images", *Agronomy*, Vol. 12, p. 2213, 2022.

51. "ZED 2 Camera Datasheet", Stereolabs Inc., San Francisco, CA.

52. Avigal, Y., J. Gao, W. Wong, K. Li, G. Pierroz, F. S. Deng, M. Theis, M. Presten

and K. Goldberg, "Simulating Polyculture Farming to Tune Automation Policies for Plant Diversity and Precision Irrigation", *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China, 2020.

53. Samiei, S., P. Rasti, J. Ly Vu, J. Buitink and D. Rousseau, "Deep learning-based Detection of Seedling Development", *Plant Methods*, Vol. 16, No. 1, p. 103, 2020.

54. Cui, Z., W. Chen and Y. Chen, "Multi-Scale Convolutional Neural Networks for Time Series Classification", ArXiv:1603.06995 [cs.CV], 2016.

55. Ismail Fawaz, H., B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. Webb, L. Idoumghar, P.-A. Muller and F. Petitjean, "InceptionTime: Finding AlexNet for time series classification", *Data Mining and Knowledge Discovery*, Vol. 34, pp. 1–27, 2020.

56. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

57. Dvornek, N., D. Yang, P. Ventola and J. Duncan, "Learning Generalizable Recurrent Neural Networks from Small Task-fMRI Datasets", *21st International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Granada, Spain, 2018.

58. Cruz-Nájera, M. A., M. G. Treviño-Berrones, M. P. Ponce-Flores, J. D. Terán-Villanueva, J. A. Castán-Rocha, S. Ibarra-Martínez, A. Santiago and J. Laria-Menchaca, "Short Time Series Forecasting: Recommended Methods and Techniques", *Symmetry*, Vol. 14, No. 6, p. 1231, 2022.

59. Le Guennec, A., S. Malinowski and R. Tavenard, "Data Augmentation for Time Series Classification using Convolutional Neural Networks", *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, Riva del Garda, Italy, 2016.

60. Box, G. E. P., G. M. Jenkins, G. C. Reinsel and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, John Wiley and Sons Inc., Hoboken, New Jersey, 2015.

61. Schwarz, G., "Estimating the Dimension of a Model", *The Annals of Statistics*, Vol. 6, No. 2, pp. 461–464, 1978.

62. Bialiev, A., "Leaf Detector Dataset", , 2023, `https://universe.roboflow.com/azim-bialiev-ui3xo/leaf-detector-phgrh\\/dataset/3`, accessed on June 12, 2024.

APPENDIX A: ROBOT HARDWARE

TarRob is a 3D gantry robot that has been developed in previous studies. During this work, both hardware and software updates of the robot were done.

A.1. TarRob Hardware

The robot has 4 stepper motors. Two of them are for X-axis movement, one for Y-axis movement, and one for Z-axis movement. The robot uses Raspberry Pi 3 as a computer and controls the stepper motors with Arduino connected to Raspberry Pi. A Wi-Fi Network was established and a GUI was developed for the robot. [16] Afterwards, the GUI was moved to Qt Creator, a database was created using MySQL, and autonomous mode was added. Additionally, full-spectrum lights and a water pump system have been added to the robot. [17] In his other ongoing thesis work, Yavuzhan made improvements to the GUI and database. He fixed the movement of the robot, which was moved and whose soil was prepared. In this study, the camera, simulation, and single-board computer were changed.

First of all, its processor has been updated from a single-board computer Raspberry to a Jetson Nano. Its camera was replaced by a ZED2 camera that was attached to the TarRob's end-effector. A 5V-4A adapter was added to the standard Jetson Nano package since with ZED 2 its standard adapter could not provide the necessary power. According to the manufacturer's specifications [51], the camera's depth-sensing characteristics are as follows:

- Depth Range: 0.3 m to 20 m
- Depth Accuracy: $< 1\%$ up to 3 m

For this research, ZED 2's specifications, are located in zed2.yaml file is:

- Resolution: 960x540 RGB Images
- Frame Rate: 20 Hertz
- Minimum Depth: 0.3 Meters
- Maximum Depth: 5.0 Meters

A.2. Intrinsic Camera Parameters

For the 960×540 resolution, intrinsic camera parameters of the ZED 2 camera are as given in Table A.1:

Table A.1. ZED 2 Camera intrinsic parameters.

Parameter	**Value**
f_x	528.272
f_y	528.272
c_x	489.542
c_y	267.570

A.3. Lens Distortion Coefficients

Even though distortion coefficients are small, the depth images were undistorted using OpenCv's corresponding method. Lens distortion coefficients of the ZED 2 camera are as given in Table A.2.

Table A.2. ZED 2 Camera distortion coefficients.

Parameter	**Value**
k_1	-0.0361431
k_2	0.00486255
p_1	-0.000249107
p_2	4.91599e-06
k_3	-0.00301238

APPENDIX B: SOFTWARE DETAILS

B.1. Gazebo Simulation of the TarRob

A Gazebo simulation of the robot already existed. In this study, the controller codes were updated, a camera model was added, and `/move2` and `/photoCommand` were added to the ROS topics to simulate image capturing before the real application. To use the Gazebo simulation, two scripts are available: one for XYZ coordinates and another for nozzle names. These `.sh` files are located in the following path: `~/TarimRobotu/src/farmbot_gazebo`

- `farmbot_xyz_starter.sh`
- `farmbot_nozzle_starter.sh`

The `farmbot_xyz_starter.sh` script includes all capabilities such as carrot model spawning, camera model, and the latest ROS topics.

B.2. Plant Growth Monitoring System

The software was developed using ROS and C++. Figure B.1 depicts ROS nodes and topics. The image capture node starts the system by giving the paths of the RGB and depth images. The software is comprised of a plant segmentation node, leaf detection node, measurements node and store features node. There is also a PCL measurements node that visualizes the results obtained by combining detection and segmentation outputs with depth data. Finally, the entire system can be triggered thanks to the dummy image captured node. The `plant_growth.sh` contains all terminal commands that enable the system to run. The system can be run with these commands, each command opens a terminal window.

(i) roscore - Start ROS core

(ii) rosrun yolo_point_cloud imageCaptured_node - This node gets the paths of the rgb and depth images and then publishes them.

(iii) rosrun yolo_point_cloud yolov8_segmentation_node.py - This node runs YOLOv8 plant detection and segmentation model.

(iv) rosrun yolo_point_cloud yolov8_leaf_detection.py - This node runs YOLOv8 leaf detection model.

(v) rosrun yolo_point_cloud measurements_node - This node runs plant phenotype computation code.

(vi) yolo_point_cloud store_features_node - This node subscribes to plant phenotypes and stores them in a csv file.

(vii) rostopic pub /image_captured std_msgs/Bool "data: true" - This command can be used as a trigger to run the system. This command is commented out.

(viii) rosrun yolo_point_cloud pcl_measurements_node - This node runs plant phenotype computation and also it shows point clouds of the bounding box and the segmentation mask. This command is commented out.

(ix) rosrun yolo_point_cloud yolov7_leaf_detection.py - This node runs YOLOv7 leaf detection model. This command is commented out.

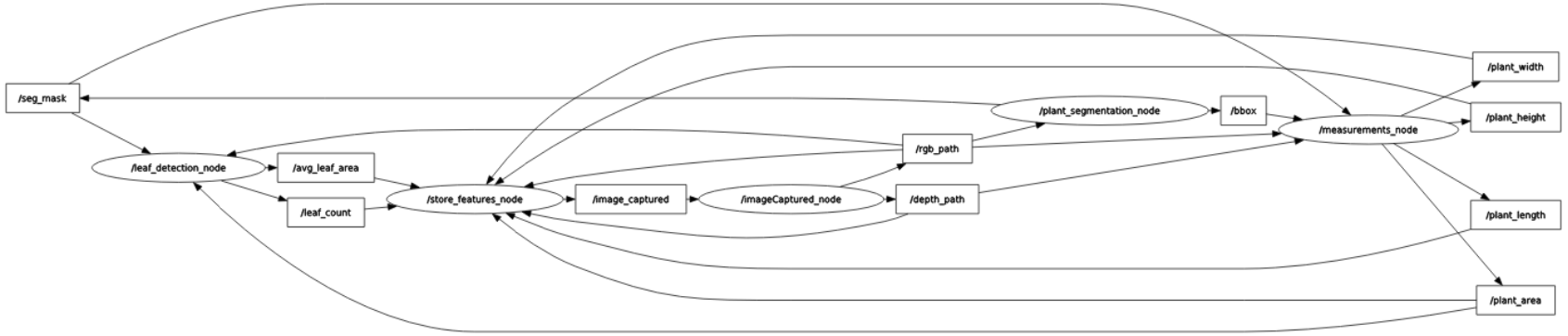


Figure B.1. ROS network of the plant growth monitoring with outputs.

B.3. Blender-Gazebo Connection

Blender's Gazebo plugin was installed, and a carrot model produced in Blender was transferred to the Gazebo environment.
Related tools can be found in: `~/TarimRobotu/Tools/BlenderTools/`

- `new_blender_export.sh`
- `random_carrot_bot.sh`

APPENDIX C: MODEL TRAINING DATASETS

This appendix contains the detailed information about the training datasets of the training of the YOLO models.

Table C.1. Plant detection dataset.

Description	**Details**
Annotated Images	800 images
Training Set	600 images
Validation Set	200 images
Training Set with Augmentation	1800 images
Augmentation Methods	Flip: Horizontal, Vertical Rotation: Between -15° and +15° Brightness: Between -15° and +15° Saturation: Between -15° and +15°
Total Annotations	2340
Average Annotations per Image	2.9

Table C.2. Leaf detection dataset.

Description	Details
Annotated Images	200 images
Training Set	150 images
Validation Set	50 images
Training Set with Augmentation	450 images
Augmentation Methods	Flip: Horizontal, Vertical Rotation: Between -15° and +15° Brightness: Between -15° and +15° Saturation: Between -15° and +15°
Total Annotations	1457
Average Annotations per Image	7.3

Table C.3. External leaf detection dataset [62].

Description	Details
Training Set	4988 images
Validation Set	475 images
Test Set	238 images
Augmentation per Training Example	3
Augmentation Methods	Flip: Horizontal, Vertical Rotation: 90° and -90° Shear: Between -15° and +15°
Total Annotations	9733
Average Annotations per Image	4.1